Rodríguez-Ferran, A., Pegon, P. and Huerta, A., Two Stress Update Algorithms for Large Strains: Accuracy Analysis and Numerical Implementation, *International Journal for Numerical Methods in Engineering*, Vol. 40, Issue 23, pp. 4363-4404, 1997

TWO STRESS UPDATE ALGORITHMS FOR LARGE STRAINS: ACCURACY ANALYSIS AND NUMERICAL IMPLEMENTATION

ANTONIO RODRÍGUEZ-FERRAN[1], PIERRE PEGON[2] AND ANTONIO HUERTA[1]*

[1]*Departamento de Matemática Aplicada III, E.T.S. de Ingenieros de Caminos, Universitat Politècnica de Catalunya, Campus Nord C-2, E-08034 Barcelona, Spain*

[2]*Structural Mechanics Unit, Institute for Systems, Informatics and Safety, Joint Research Centre of the European Commission, I-21020 Ispra (Va), Italy*

ABSTRACT

Two algorithms for the stress update (i.e., time integration of the constitutive equation) in large-strain solid mechanics are compared from an analytical point of view. The order of the truncation error associated to the numerical integration is deduced for each algorithm *a priori*, using standard numerical analysis. This accuracy analysis has been performed by means of a convected frame formalism, which also allows a unified derivation of both algorithms in spite of their inherent differences. Then the two algorithms are adapted from convected frames to a fixed Cartesian frame and implemented in a small-strain finite element code. The implementation is validated by means of a set of simple deformation paths (simple shear, extension, extension and compression, extension and rotation) and two benchmark tests in non-linear mechanics (the necking of a circular bar and a shell under ring loads). In these numerical tests, the observed order of convergence is in very good agreement with the theoretical order of convergence, thus corroborating the accuracy analysis.

No. of Figures: 28. No. of Tables: 1. No. of References: 21.

KEY WORDS: large strains; stress update; error analysis; convected frames; nonlinear computational mechanics; finite element method

1. INTRODUCTION

Two basic types of non-linearity are encountered in non-linear solid mechanics: material (inelastic constitutive behaviour) and geometric (large strains). Non-linear material behaviour is typically described by a rate-form constitutive equation, relating the rate of stress to the rate of strain and some internal variables.[1] If deformations are large, then the changing configuration must be accounted for when stating and handling the rate-form constitutive equation.

* Correspondence to: Antonio Huerta: Departamento de Matemática Aplicada III, E.T.S. de Ingenieros de Caminos, Universitat Politècnica de Catalunya, Campus Nord C-2, E-08034 Barcelona, Spain. E-mail: huerta@etseccpb.upc.es

Contract grant sponsor: European Commission; Contract grant number: ERB-4050-PL-94-1116
Contract grant sponsor: Ministerio de Educacion y Ciencia; Contract grant number: DGICYT PB94-1200

In this paper, the accuracy analysis of two stress update algorithms for the numerical time integration of hypoelastic laws in large strain analysis is shown. That is, the order of the time discretization error associated to each algorithm is evaluated by means of standard tools in numerical analysis. Therefore, an *a priori* knowledge of the accuracy for each algorithm is provided.

The accuracy analysis is enabled by the use of a convected frame formalism. Convected frames are attached to the body and deform with it, thus, the statement and time integration of the constitutive equation becomes a straightforward task. In fact, if convected frames are chosen as reference, any standard finite difference scheme can be employed for the numerical time integration of the constitutive equation, and the truncation error can be determined in the usual way.

The use of convected frames has also allowed a complete derivation of both algorithms from a general and unified perspective, in spite of the important differences inherent to both methods. The two methods have been previously presented in a fixed frame setting.[2, 3] The first algorithm,[2] uses the full-strain tensor including quadratic terms to account for large strains, and it is first-order-accurate in time. The second algorithm[3] employs the same strain measure as in small strain analysis but computed in the midstep configuration. It is second-order-accurate in time.

The convected frame formalism may be employed to develop a large-strain finite element code.[4] However, the standard approach in non-linear computational mechanics is the use of a fixed Cartesian frame.[5] It leads to a simpler description of motion (because the frame does not change) but, on the other hand, the integration of constitutive laws becomes more involved. Because most existing large strain codes employ a Cartesian frame, just like for small strains, the two algorithms are adapted from convected frames to a fixed frame. This allows to recover the classical form of both algorithms which can be found in References 6 and 7, and to recognize them as the algorithms originally presented in References 2 and 3.

Various implementation aspects for both algorithms are commented. It is shown, in particular, that very few additional features must be added to a code with small-strain and non-linear material behaviour to enable its use for large-strain analysis.

The two algorithms are tested and compared with the help of various simple deformation paths and two benchmark tests. The tests are performed with different values of the time increment to assess its influence on the results. Both algorithms behave in good agreement with the *a priori* accuracy analysis.

This paper is organized as follows. The basic notions of large-strain solid mechanics in a convected frame context are reviewed in Section 2. Section 3 deals with the two stress update algorithms. First, the two algorithms are presented in Section 3.1. The error analysis is then developed in Sections 3.2 and 3.3. Some computational aspects are treated in Section 4. Three topics are covered: the adaptation of the algorithms from convected frames to a fixed Cartesian frame (Section 4.1); the implementation in a small-strain code (Section 4.2); the computational cost (Section 4.3). Various numerical examples can be found in Section 5. With the help of simple deformation paths the relative performance of the two algorithms is assessed. These tests corroborate the theoretical results of Section 3. Then, two well-known benchmark tests, a necking analysis and a shell under ring loads, confirm the expected results for continuum and structural elements. Finally, some concluding remarks are made in Section 6.

2. LARGE-STRAIN SOLID MECHANICS IN CONVECTED FRAMES

2.1. Introductory remarks

A common approach in continuum mechanics is to use a fixed orthonormal frame, denoting each spatial point by its Cartesian co-ordinates with respect to that frame. It is also possible, however, to employ convected material co-ordinates. By doing so, every material particle is identified, throughout the deformation process, by the same material co-ordinates, which are convected by the body motion. The reference is then a field of material convected frames, associated to material co-ordinates at every point.

This represents a fully Lagrangian approach: not only the description focuses on the motion of material particles, but also on a reference field of material convected frames which deform with the body. These convected frames may never be orthonormal. Moreover, according to Truesdell and Toupin (Reference 8, Section 66), "*the mathematical difficulties which accompany the material description have limited its use to two special ends*: *problems in one dimension and the proof of general theorems.*" However, in exchange for these *difficulties* (which are minor, since only a few, very basic concepts in differential geometry are required), the use of convected frames considerably simplifies both the statement and the numerical time integration of constitutive equations for large strain solid mechanics. For this reason, a convected frame formalism will be employed here to present the two stress update algorithms.

2.2. Kinematics

Consider the motion of a deformable body Ω through the usual Euclidean space $\mathbb{R}^3$. The position of its particles is referred to a fixed orthonormal frame $(O, \mathbf{e}_\alpha)$, which is the frame of the observer, see Figure 1. Let $\mathbf{Op} = z^\alpha \mathbf{e}_\alpha$ be the current position vector at time t of the material particle initially located at $\mathbf{OP} = Z^\alpha \mathbf{e}_\alpha$, where the convention of summation on repeated indices is adopted.

In order to take into account the material continuum and discontinuous deformations, the points may be related to an embedded mapping manifold $C(\Omega)$ of curvilinear material co-ordinates x^i. Each material particle M is identified by three material co-ordinates (x^1, x^2, x^3) which are convected

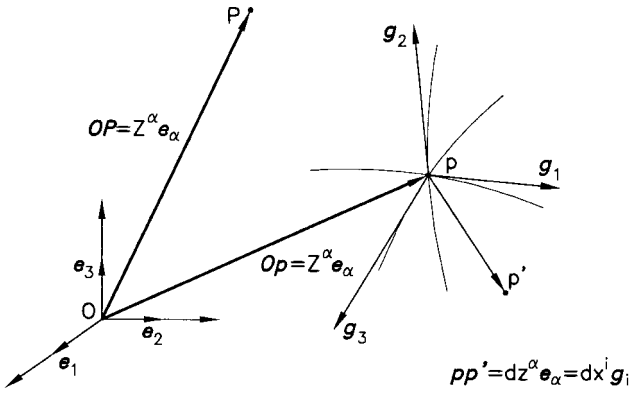


Figure 1. Fixed and material reference frames

by the body motion, thus remaining "attached" to the particle. For instance, a possible choice of material co-ordinates is the initial orthonormal co-ordinates, $x^i = Z^i$.

The motion of the body is expressed by

$$z^\alpha = z^\alpha(x^i, t) \tag{1}$$

which gives the orthonormal co-ordinates z^α at time t of the particle with material co-ordinates x^i. Throughout the paper, Latin indices are employed for material components and Greek indices for orthonormal components.

To present the basic features of convected frames, let us focus on two neighbouring material points. At the current time t, consider the material point $M(x^i)$ located at the geometrical point p. Any geometrical point p' in the vicinity of p may be referenced to the orthonormal frame by

$$\mathbf{pp}' = \mathrm{d}z^\alpha \mathbf{e}_\alpha \tag{2}$$

On the other hand, point p' may also be considered the location of the material point $M'(x^i + \mathrm{d}x^i)$; see Figure 1. Then, the basis vectors $\mathbf{g}_i$ of the natural reference frame at the material point $M(x^i)$ are implicitly defined by

$$\mathbf{pp}' = \mathrm{d}x^i \mathbf{g}_i \tag{3}$$

Combining equations (1)–(3), the relation between the vectors $\mathbf{e}_\alpha$ of the orthonormal frame and the vectors $\mathbf{g}_i$ of the natural reference frame field is easily derived as

$$\mathbf{g}_i(x^j, t) = \frac{\partial z^\alpha(x^j, t)}{\partial x^i} \mathbf{e}_\alpha \tag{4}$$

For the particular case of $x^i = Z^i$ (initial orthonormal co-ordinates taken as material co-ordinates), equation (4) becomes

$$\mathbf{g}_i(Z^j, t) = F_i^\alpha(Z^j, t) \mathbf{e}_\alpha \tag{5}$$

where $F_i^\alpha = \partial z^\alpha / \partial Z^i$ are the components of the standard deformation gradient $\mathbf{F}$, see Reference 1.

Starting from the definition of the motion, equation (1), and the expression of vector $\mathbf{Op}$, the current velocity of the material point $M(x^i)$ is obtained after partial derivation with respect to time as

$$\mathbf{v}(x^i, t) = \frac{\partial z^\alpha(x^i, t)}{\partial t} \mathbf{e}_\alpha \tag{6}$$

2.3. *The dragging-along process*

The comparison between the definitions of $\mathbf{g}_i$ and $\mathbf{v}$ given, respectively, at equations (4) and (6) allows to derive the fundamental relation

$$\frac{\partial \mathbf{g}_i}{\partial t} = \frac{\partial \mathbf{v}}{\partial x^i} \tag{7}$$

A graphical interpretation of this relation is shown in Figure 2. For simplicity, all material co-ordinates x^j are fixed except x^i. Two neighbouring material points, $M(x^i)$ and $M'(x^i + \mathrm{d}x^i)$, are under consideration:

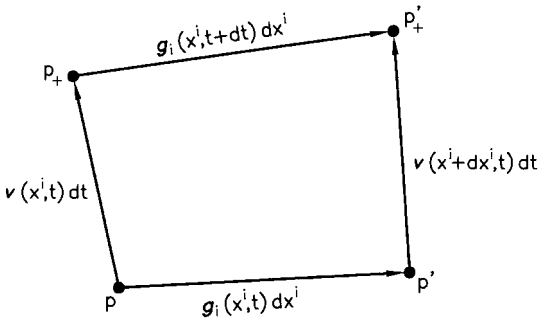


Figure 2. Dragging-along process of the reference frame field

(i) At current time t, they occupy the geometrical locations $p(x^i,t)$ and $p'(x^i+\mathrm{d}x^i,t)$, and $\|\mathbf{pp}'\| = \|\mathbf{g}_i(t)\,\mathrm{d}x^i\|$.

(ii) At time $t+\mathrm{d}t$, they occupy the locations $p_+(x^i,t+\mathrm{d}t)$ and $p'_+(x^i+\mathrm{d}x^i,t+\mathrm{d}t)$, with $\|\mathbf{pp}_+\| = \|\mathbf{v}\,\mathrm{d}t\|$ and $\|\mathbf{p}_+\mathbf{p}'_+\| = \|\mathbf{g}_i(t+\mathrm{d}t)\,\mathrm{d}x^i\|$.

It can be seen in Figure 2 how the vector field $\mathbf{v}\,\mathrm{d}t$, which transforms p into p_+ and p' into p'_+, also transforms the vector $\mathbf{g}_i(x^j,t)$ into the vector $\mathbf{g}_i(x^j,t+\mathrm{d}t)$. In other words, $\mathbf{g}_i$ is dragged along (or convected) by the vector field $\mathbf{v}\,\mathrm{d}t$, hence by the motion itself. Therefore, the natural reference frame is convected by the body motion.

The main consequence, underlined by Oldroyd (Reference 9, Section 1), is the following: "*the device of representing all tensor quantities (say* $\mathbf{A}$*) associated with the material by their convected components (say* A^{ij}*) allows similar quantities associated with the same material point at different times to be added, component by component, to give a similar tensor quantity as the sum.*"

For instance, in the current reference frame field $(M,\mathbf{g}_i)$, the quantities $\int_{t_1}^{t} A^{ij}(t')\,\mathrm{d}t'$ and $\partial A^{ij}/\partial t$ are still components of a tensor field obtained merely by adding or subtracting components A^{ij} at different instants. This situation is in sharp contrast with the one encountered with a fixed orthonormal frame. If $\mathbf{A}$ is represented by its orthonormal components $A^{\alpha\beta}$, then $\int_{t_1}^{t} A^{\alpha\beta}(t')\,\mathrm{d}t'$ and $\partial A^{\alpha\beta}/\partial t$ are *not* components of a tensor field. The basic idea is that $A^{\alpha\beta}(t_1)$ and $A^{\alpha\beta}(t_2)$ are referred to different configurations, and they must be transformed into a common configuration before they can be added or subtracted.[3]

This different behaviour makes material co-ordinates especially attractive for the development and numerical treatment of integro-differential constitutive relationships, which must be *objective* (that is, meaningful for the body, but independent from the observer's viewpoint). Indeed, by employing material co-ordinates which are convected by body motion itself, all the discussions about the verification of the *principle of material frame indifference* (see Reference 10, Sections 19 and 19A), also called *objectivity principle*, *a priori* disappear.

Regarding the numerical treatment, any rate-form constitutive equation may be integrated in time using any standard finite difference scheme, because all the quantities appearing in these schemes have a tensorial meaning related to the body and can be directly combined, as pointed out above.

2.4. *The strain and rate-of-deformation tensors*

Once kinematics and the dragging-along process have been presented, the next required ingredient is a measure of strain and strain rate. The starting point is the current separation $\mathrm{d}s$ between

two neighbouring material points $M(x^i)$ and $M'(x^i + \mathrm{d}x^i)$. Combining equations (2) and (3), $\mathrm{d}s$ can be represented as

$$(\mathrm{d}s)^2 = \mathbf{pp}' \cdot \mathbf{pp}' = \mathrm{d}z^\alpha \, \mathrm{d}z^\alpha = (\mathbf{g}_i \cdot \mathbf{g}_j) \, \mathrm{d}x^i \, \mathrm{d}x^j = g_{ij} \, \mathrm{d}x^i \, \mathrm{d}x^j \tag{8}$$

where g_{ij} are the covariant components of the metric tensor field $\mathbf{G}$ in the convected frame. An important scalar associated to $\mathbf{G}$ is the ratio of spatial unit volume to material unit volume $\sqrt{g}$,[4] given by

$$\sqrt{g} = \sqrt{\det(g_{ij})} \tag{9}$$

Both $\mathbf{G}$ and $\sqrt{g}$ contain information about the current configuration of the body. Note, for instance, that if the initial orthonormal co-ordinates are taken as material co-ordinates ($x^i = Z^i$), then in the initial configuration $\mathbf{G}$ is the identity tensor and $\sqrt{g}$ is one, thus reflecting an undeformed state.

The deformation between the initial time $t = 0$, which is chosen as a reference, and the current time t can be obtained by comparing the two separations $\mathrm{d}s(t)$ and $\mathrm{d}s(0)$. Recalling the definition of the metric tensor, equation (8), the difference between the two separations is expressed as

$$[\mathrm{d}s(t)]^2 - [\mathrm{d}s(0)]^2 = [g_{ij}(t) - g_{ij}(0)] \, \mathrm{d}x^i \, \mathrm{d}x^j = 2\varepsilon_{ij}(t) \, \mathrm{d}x^i \, \mathrm{d}x^j \tag{10}$$

Thanks to the dragging-along process, see Figure 2, $g_{ij}(0)$ is a quantity memorized at time $t = 0$ but which still has a tensorial meaning at time t, so it can be subtracted from $g_{ij}(t)$ to define the components of a meaningful strain tensor $\boldsymbol{\varepsilon}$, which is in fact the well-known Almansi–Euler strain tensor.

The rate-of-deformation tensor $\mathbf{d}$ can be defined in the current reference frame field $(M, \mathbf{g}_i)$ simply by taking the time derivative of $\varepsilon_{ij}(t)$,

$$d_{ij}(t) = \frac{\partial \varepsilon_{ij}(t)}{\partial t} = \frac{1}{2} \frac{\partial g_{ij}(t)}{\partial t} \tag{11}$$

By using the basic properties of covariant differentiation, see Reference 4, it can be shown that $\mathbf{d}$ is indeed the symmetric part of the velocity gradient in the convected frame. If such definition of $\mathbf{d}$ is adopted, it is possible to define the strain tensor $\boldsymbol{\varepsilon}$ *a posteriori* according to

$$\varepsilon_{ij}(t) = \int_0^t d_{ij}(t') \, \mathrm{d}t' \tag{12}$$

Note that, as commented in the previous subsection, the use of material components to represent tensors has enabled a straightforward definition of strain and strain rate measures, equations (10)–(12).

2.5. *The stress tensor*

The next essential ingredient is a stress measure. To describe the current stress state at the material point $M(x^i)$, the Cauchy analysis in curvilinear co-ordinates (see Reference 11, Section 10.3) leads to the introduction of the *relative* stress tensor $\boldsymbol{\sigma}(M, t)$. In the fixed orthonormal frame, the components $\sigma^{\alpha\beta}$ of $\boldsymbol{\sigma}$ are those of the classical Cauchy stress tensor. The material components σ^{ij} are related to $\sigma^{\alpha\beta}$ by the transformation rule

$$\sigma^{ij} = \sqrt{g} \frac{\partial x^i}{\partial z^\alpha} \frac{\partial x^j}{\partial z^\beta} \sigma^{\alpha\beta} \tag{13}$$

2.6. *The governing equations*

For a wide range of problems in solid mechanics, it is a common assumption that (i) mechanical and thermal effects are uncoupled and (ii) the density is a constant. The mechanical problem then involves the momentum balance and the constitutive equation.

The weak form of the momentum balance is the so-called balance of virtual power, which equates the virtual power of internal forces and the virtual power of external forces.[12, 5] These virtual powers are global quantities, obtained through integration over the whole domain. The convected material approach allows to merge two attractive features: dealing with *current* quantities associated with the *current* state of the body Ω and, at the same time, being attached to a *time-invariant* mapping $C(\Omega)$ of material co-ordinates. The integration domain is therefore $C(\Omega)$ throughout the whole body motion.

As for the constitutive equation, it is typically written in rate form, relating a stress rate to a certain measure of strain rate.[1] If material co-ordinates are employed, a stress rate tensor can be defined in the current reference frame field $(M, \mathbf{g}_i)$ simply by taking the time derivative of σ^{ij},

$$\mathscr{T}(\sigma)^{ij} = \frac{\partial \sigma^{ij}}{\partial t} \tag{14}$$

As commented in Reference 13, Section 55bis, this is the Truesdell stress rate, which has a very simple expression in material components, see equation (14). This particular rate was introduced because, very physically, during the dragging-along process of the stress it gives priority to the resultant force *vector* rather than to the stress *tensor* itself (see also Reference 14, Section 1). Many other stress rates have been employed in non-linear solid mechanics, but a review of the various choices is beyond the scope of this work.

The Truesdell rate can be employed to write a general non-linear rate-form constitutive equation as

$$\mathscr{T}(\sigma)^{ij} = A^{ij}(\boldsymbol{\sigma}, \mathbf{v}) \tag{15}$$

where $\mathbf{A}$ may depend both on the strain rate (symbolically represented in equation (15) by velocity $\mathbf{v}$) and the state of stress (and, eventually, some internal variables). A particular case of this general expression is the hypoelastic constitutive law

$$\mathscr{T}(\sigma)^{ij} = C^{ijkl} d_{kl} \tag{16}$$

which states the linear dependence of the Truesdell rate on the rate-of-deformation tensor. C^{ijkl} are the material components of the fourth-order elastic modulus tensor.

As commented previously, the use of convected frames guarantees *a priori* the objectivity of the constitutive equation (15), because it is stated in terms of material components (i.e. associated to the body and not to the observer's frame). Numerical time integration is also straightforward. The stress rate can be discretized, by means of a simple finite difference formula, into

$$\mathscr{T}(\sigma)^{ij}(t_{n+\theta}) \approx \frac{\sigma^{ij}(t_{n+1}) - \sigma^{ij}(t_n)}{\Delta t} \tag{17}$$

where $t_{n+\theta}$ is an intermediate instant between t_n and $t_{n+1} = t_n + \Delta t$. Again, the use of material components is essential to allow $\sigma^{ij}(t_{n+1})$ and $\sigma^{ij}(t_n)$ to be subtracted and yield meaningful tensorial quantities. This is not the situation if a fixed orthonormal frame is employed: as shown

in Section 4.1 and in References 6 and 7, $\sigma^{\alpha\beta}(t_{n+1})$ and $\sigma^{\alpha\beta}(t_n)$ have to be transformed into a common configuration before they can be subtracted, thus making the time integration of the constitutive equation more elaborate.

3. TWO STRESS UPDATE ALGORITHMS FOR LARGE STRAINS

The equilibrium equation for quasistatic processes is typically solved with an implicit, incremental approach.[15, 16] The fundamental unknowns are then the incremental displacements $\Delta\mathbf{u} = \Delta z^{\alpha}\mathbf{e}_{\alpha}$ from one equilibrium configuration of the body at time t_n to a new equilibrium configuration at time t_{n+1}. These incremental displacements are first predicted and then iteratively corrected. Within each iteration, the constitutive equation must be integrated over the time increment to update the stresses from t_n to t_{n+1} and check the equilibrium.

3.1. Two stress update algorithms

Two algorithms for the numerical time integration of the constitutive equation will be presented in this subsection. An intuitive approach will be made. First, a restriction will be imposed on the problem, to get a particular version of the algorithms in a straightforward manner. Then the restriction will be removed, and the general expressions of the algorithms will be obtained. A more rigorous course will be followed in the next subsection, where an error analysis for both algorithms is performed.

Profiting from the simple expression of the Truesdell rate in material components, equation (14), the rate-form constitutive equation (15) can be integrated over the time increment. For the hypo-elastic equation (16), for instance, it yields

$$\int_{t_n}^{t_{n+1}} \frac{\partial \sigma^{ij}(t)}{\partial t}\,\mathrm{d}t = {}^{n+1}\sigma^{ij} - {}^{n}\sigma^{ij} = \int_{t_n}^{t_{n+1}} C^{ijkl}(t) d_{kl}(t)\,\mathrm{d}t \tag{18}$$

where left superscripts denote time (i.e. ${}^{n}\sigma^{ij} \equiv \sigma^{ij}(t_n)$).

Assume for the moment that the material components C^{ijkl} are constant. With this simplifying restriction, the RHS of equation (18) can be put, recalling equation (12), as

$$C^{ijkl}\int_{t_n}^{t_{n+1}} d_{kl}(t)\,\mathrm{d}t = C^{ijkl}[{}^{n+1}\varepsilon_{kl} - {}^{n}\varepsilon_{kl}] = C^{ijkl}\Delta\varepsilon_{kl} \tag{19}$$

where $\Delta\varepsilon_{kl}$ are the material components of the increment of strain tensor. Recalling the definition of the strain tensor in equation (10), the strain increment $\Delta\varepsilon_{kl}$ can be obtained as

$$\Delta\varepsilon_{kl} = \frac{1}{2}[{}^{n+1}g_{kl} - {}^{n}g_{kl}] = \frac{1}{2}\left[\frac{\partial^{n+1}z^{\alpha}}{\partial x^{k}}\frac{\partial^{n+1}z^{\alpha}}{\partial x^{l}} - \frac{\partial^{n}z^{\alpha}}{\partial x^{k}}\frac{\partial^{n}z^{\alpha}}{\partial x^{l}}\right] \tag{20}$$

Remark 1. The increment of strain $\Delta\varepsilon_{kl}$ is a purely kinematical quantity, independent of material behaviour.

Remark 2. Equation (20) allows to compute $\Delta\varepsilon_{kl}$ from the orthonormal co-ordinates ${}^{n}z^{\alpha}$ and ${}^{n+1}z^{\alpha}$, which are available in the incremental analysis. No assumption about the body motion between t_n and t_{n+1} is required.

Combining equations (18) and (19), a stress update algorithm can be written as

$$^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij} + C^{ijkl}\Delta\varepsilon_{kl} \tag{21}$$

with $\Delta\varepsilon_{kl}$ computed according to equation (20). It must be remarked that, in this particular case of constant material components C^{ijkl}, the algorithm given by equation (21) *has no time-discretization error*, because it has been obtained through exact time integration of the constitutive equation, with no need of numerical time integration.

In many cases, however, the material components of $\mathbf{C}$ are not constant. A common assumption in computational mechanics is that of constant *orthonormal* components $C^{\alpha\beta\gamma\delta}$, written in terms of the Lamé constants λ and μ as

$$C^{\alpha\beta\gamma\delta} = \lambda\delta_{\alpha\beta}\delta_{\gamma\delta} + \mu(\delta_{\alpha\gamma}\delta_{\beta\delta} + \delta_{\alpha\delta}\delta_{\beta\gamma}) \tag{22}$$

where $\delta_{\alpha\beta}$ is the Kronecker delta. If the orthonormal components $C^{\alpha\beta\gamma\delta}$ are constant, the material components C^{ijkl} are not. This means that the stress update algorithm (21) must be modified by specifying when C^{ijkl} is evaluated. Two possible choices, resulting in two stress update algorithms, will be shown next.

First algorithm

The first choice is to evaluate C^{ijkl} at the beginning of the increment, t_n. The increment of strain $\Delta\varepsilon_{kl}$ presented in equation (20) is written in an *equivalent* form, using only quantities at t_n, after substitution of $^{n+1}z^{\alpha} = {}^{n}z^{\alpha} + \Delta z^{\alpha}$ in equation (20). The stress algorithm is then

$$^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij} + {}^{n}C^{ijkl}\Delta\varepsilon_{kl} \tag{23a}$$

$$\Delta\varepsilon_{kl} = \frac{1}{2}\left[\frac{\partial\Delta z^{\alpha}}{\partial x^{k}}\frac{\partial({}^{n}z^{\alpha})}{\partial x^{l}} + \frac{\partial({}^{n}z^{\alpha})}{\partial x^{k}}\frac{\partial\Delta z^{\alpha}}{\partial x^{l}} + \frac{\partial\Delta z^{\alpha}}{\partial x^{k}}\frac{\partial\Delta z^{\alpha}}{\partial x^{l}}\right] \tag{23b}$$

Remark 3. The expressions of $\Delta\varepsilon_{kl}$ in equations (20) and (23b) are *equivalent*, so the strain increment employed in the first algorithm is still the *exact* one.

Remark 4. However, the stress update algorithm is no longer exact. Contrary to what happened in the particular case of constant C^{ijkl}, equation (21), the last term in equation (23a) is not the exact value of the stress increment, but an approximation affected by a time-discretization error. An error analysis will be performed in the next subsection.

Second algorithm

An intuitively better approach is a time-centred scheme, where quantities are evaluated at the midstep instant $t_{n+1/2}$. However, since an incremental strategy has been adopted, the only available kinematical data are the orthonormal co-ordinates at the beginning and the end of the time step. Nothing is known about the body motion between t_n and t_{n+1}. As a consequence, an hypothesis is needed to evaluate quantities at $t_{n+1/2}$. A common choice is to assume that the velocity is *constant* within the time step, $\bar{v}^{\alpha} = \Delta z^{\alpha}/\Delta t$. The midstep configuration is then

$$^{n+1/2}\bar{z}^{\alpha} = {}^{n}z^{\alpha} + \tfrac{1}{2}\Delta z^{\alpha} = {}^{n+1}z^{\alpha} - \tfrac{1}{2}\Delta z^{\alpha} \tag{24}$$

The bar is used to emphasize that the expressions of $\bar{v}^\alpha$ and ${}^{n+1/2}\bar{z}^\alpha$ are based on the constant-velocity assumption. Equation (24) allows to express the strain increment given by equation (20) in an *equivalent* form, employing midstep orthonormal co-ordinates. The stress update algorithm is

$$ {}^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij} + {}^{n+1/2}\overline{C}^{ijkl}\Delta\varepsilon_{kl} \tag{25a} $$

$$ \Delta\varepsilon_{kl} = \frac{1}{2}\left[\frac{\partial^{n+1/2}\bar{z}^\alpha}{\partial x^k}\frac{\partial(\Delta z^\alpha)}{\partial x^l} + \frac{\partial(\Delta z^\alpha)}{\partial x^k}\frac{\partial^{n+1/2}\bar{z}^\alpha}{\partial x^l}\right] \tag{25b} $$

where ${}^{n+1/2}\overline{C}^{ijkl}$ are the material components of the modulus tensor computed at the midstep under the constant-velocity assumption.

Remark 5. The constant-velocity assumption is just a computational convenience that allows to compute the *exact* strain increment in terms of midstep quantities, equation (25b). In fact, no restriction is placed on the *true* body motion between t_n and t_{n+1} and, similarly to Remark 3, the expressions of $\Delta\varepsilon_{kl}$ in equations (20) and (25b) are *equivalent*.

Remark 6. The stress update algorithm given by equations (25), however, is not exact. As remarked for the first algorithm, the last term in equation (25a) is not the exact stress increment, but only an approximation.

Remark 7. The two stress update algorithms, equations (23) and (25), are *identical* if the material components of the modulus tensor are constant, equation (21). They differ, however, for variable C^{ijkl}. The error analysis of the next subsection shows that the time-centred strategy of the second algorithm is indeed more accurate than the forward scheme of the first algorithm.

The two algorithms just presented are *incrementally objective* (in the sense of References 17 and 18), i.e., they treat rigid rotations correctly. Indeed, if the body undergoes a rigid rotation, the separation between its particles does not change, and this results in a null strain increment, equation (10). Since the two algorithms employ the exact strain increment, equations (23b) and (25b), they both correctly predict no variation in the material stress components, ${}^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij}$.

Incremental objectivity is often presented as the discrete counterpart of the principle of objectivity, see References 17 and 18, which states that the constitutive equations must be independent of the observer's choice regarding the reference frame. It must be remarked, however, that a null strain increment $\Delta\varepsilon_{kl}$ does *not* necessarily imply that the motion between t_n and t_{n+1} is a rigid rotation. Many other paths are possible, including such simple ones as the parallel translation of all the material points, depicted in Figure 3, which may cause loading/unloading effects. As a consequence, *assuming* a rigid rotation and demanding the numerical algorithm to perform in accordance with that assumption is an observer's choice. Although this choice may be very reasonable, it is inherently non-objective (a different assumption would lead to a different requirement on the numerical algorithm). For this reason, the term *priority on rigid motion* is preferred by the authors over *incremental objectivity*.

Moreover, the treatment of rigid rotations is not the only concern. Although both algorithms behave identically for this particular test, the error analysis and the numerical examples show that the general behaviour is quite different. These differences are in accuracy, on the numerical implementation and, also, on the response to several non-rigid deformation paths different from simple rigid rotation. Thus, the performance of large strain algorithms must be evaluated beyond rigid motion into a more general error analysis.

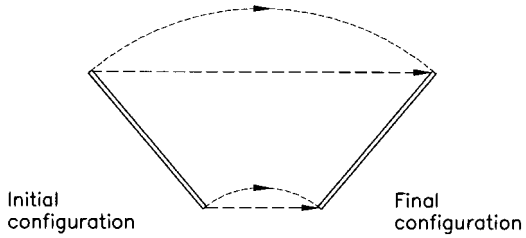


Figure 3. Two possible paths from an initial to a final configuration

It is important to note that both algorithms can be adapted to handle rate-form elastoplastic constitutive equations. As commented in Reference 3, the key idea is to profit the additive decomposition of the rate-of-deformation tensor **d** into elastic and plastic parts to perform an operator split of the constitutive equation.

3.2. Error analysis of the algorithms

The two algorithms are compared with the help of various numerical tests in Section 5. The basic goal of the comparison is to show with numerical experiments the superior performance of the second algorithm. A theoretical justification of such behaviour is presented in this subsection, where the error associated to the time discretization is analysed. The analysis shows that the second algorithm is second-order-accurate in time, while the first one is only first-order-accurate.

To perform the error analysis, a more rigorous approach than in the presentation of the algorithms is followed, accounting from the start with variable material components C^{ijkl}. If finite differences are employed to discretize the hypoelastic equation (16), the midpoint rule renders

$$\frac{\partial \sigma^{ij}}{\partial t}(t_{n+1/2}) = {}^{n+1/2}C^{ijkl}\,{}^{n+1/2}d_{kl} = \frac{{}^{n+1}\sigma^{ij} - {}^{n}\sigma^{ij}}{\Delta t} + \mathcal{O}(\Delta t^2) \tag{26}$$

which can be rearranged into

$${}^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij} + {}^{n+1/2}C^{ijkl}(\Delta t^{n+1/2} d_{kl}) + \mathcal{O}(\Delta t^3) \tag{27}$$

The constant-velocity assumption has not been introduced in equation (27), so ${}^{n+1/2}C^{ijkl}$ and ${}^{n+1/2}d_{kl}$ are the true midstep values. It is shown in the next subsection that $\Delta t\,{}^{n+1/2}d_{kl}$ is a third-order approximation to the strain increment $\Delta\varepsilon_{kl}$,

$$\Delta\varepsilon_{kl} = \Delta t\,{}^{n+1/2}d_{kl} + \mathcal{O}(\Delta t^3) \tag{28}$$

Note that, since ${}^{n+1/2}d_{kl}$ is an instantaneous quantity which does not depend on Δt, equation (28) also states that the strain increment is $\mathcal{O}(\Delta t)$.

Substituting equation (28) into equation (27) gives

$${}^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij} + {}^{n+1/2}C^{ijkl}\Delta\varepsilon_{kl} + \mathcal{O}(\Delta t^3) \tag{29}$$

Apart from the error term, which was not accounted for in the presentation of the algorithms, the only difference between equation (29) and the algorithms, equations (23) and (25), is in the material components of the modulus tensor. The error analysis is completed by studying the

effect of replacing ${}^{n+1/2}C^{ijkl}$ in equation (29) by ${}^{n}C^{ijkl}$ (first algorithm) or by ${}^{n+1/2}\overline{C}^{ijkl}$ (second algorithm).

First-order algorithm

A first-order Taylor's expansion of C^{ijkl} around t_n shows that

$$ {}^{n+1/2}C^{ijkl} = {}^{n}C^{ijkl} + \mathcal{O}(\Delta t) \tag{30} $$

where it is assumed that the time derivative of C^{ijkl} exists and is bounded. Combining equations (28)–(30) results in

$$ {}^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij} + {}^{n}C^{ijkl}\Delta\varepsilon_{kl} + \mathcal{O}(\Delta t^2) \tag{31} $$

Equation (31) only differs from the first stress update algorithm, equation (23), on the error term. Since this error is order 2 for one time step (from t_n to t_{n+1}), the global error is order 1. In conclusion, *the first algorithm is first-order-accurate in time.*

Second-order algorithm

As shown in the appendix, the constant-velocity assumption provides second-order approximation in the midstep material components of the modulus tensor,

$$ {}^{n+1/2}C^{ijkl} = {}^{n+1/2}\overline{C}^{ijkl} + \mathcal{O}(\Delta t^2) \tag{32} $$

Substituting equation (32) into equation (29), and employing the expression of $\Delta\varepsilon_{kl}$ in equation (28) yields

$$ {}^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij} + {}^{n+1/2}\overline{C}^{ijkl}\Delta\varepsilon_{kl} + \mathcal{O}(\Delta t^3) \tag{33} $$

which, except for the error term, coincides with the second algorithm, equation (25). Since the error is order 3 for one time step, the global error is order 2. In conclusion, *the second algorithm is second-order-accurate in time.*

3.3. The strain increment and the midpoint rule

The superiority of a time-centred approach (second-order algorithm) over a forward scheme (first-order algorithm) is illustrated by the error analysis just presented. As a further justification, it is shown here that, if a generalized trapezoidal rule is employed to integrate the hypoelastic equation (16),

$$ {}^{n+1}\sigma^{ij} = {}^{n}\sigma^{ij} + {}^{n+\theta}C^{ijkl}(\Delta t\, {}^{n+\theta}d_{kl}), \quad 0 \leqslant \theta \leqslant 1 \tag{34} $$

the midpoint rule ($\theta = \frac{1}{2}$), in combination with the constant-velocity assumption, is the only one which works with the *exact* strain increment.

The material components of the rate-of-deformation tensor defined in equation (11) can be put more explicitly, considering equations (4), (7) and (8), as

$$ d_{kl}(t) = \frac{1}{2}\left[\frac{\partial z^{\alpha}(t)}{\partial x^{k}}\frac{\partial v^{\alpha}(t)}{\partial x^{l}} + \frac{\partial v^{\alpha}(t)}{\partial x^{k}}\frac{\partial z^{\alpha}(t)}{\partial x^{l}}\right] \tag{35} $$

If the velocity is assumed constant within the time step, $\bar{v}^\alpha = \Delta z^\alpha / \Delta t$, the body motion from t_n to t_{n+1} is represented by

$$^{n+\theta}\bar{z}^\alpha = {}^n z^\alpha + \theta \Delta z^\alpha, \quad 0 \leqslant \theta \leqslant 1 \tag{36}$$

By modifying equation (35) according to equation (36), the approximation to the rate-of-deformation based on the constant-velocity assumption can be written as

$$^{n+\theta}\bar{d}_{kl} = \frac{1}{2}\frac{1}{\Delta t}\left[\frac{\partial {}^n z^\alpha}{\partial x^k}\frac{\partial(\Delta z^\alpha)}{\partial x^l} + \frac{\partial(\Delta z^\alpha)}{\partial x^k}\frac{\partial {}^n z^\alpha}{\partial x^l} + 2\theta\frac{\partial(\Delta z^\alpha)}{\partial x^k}\frac{\partial(\Delta z^\alpha)}{\partial x^l}\right] \tag{37}$$

Comparing equation (37) to the *exact* expression of the strain increment employed in equation (23b) gives

$$\Delta\varepsilon_{kl} = \Delta t\, {}^{n+\theta}\bar{d}_{kl} \;\Leftrightarrow\; \theta = \tfrac{1}{2}. \tag{38}$$

Since $\Delta t\, {}^{n+\theta}\bar{d}_{kl}$ is employed as an approximation to the strain increment in the trapezoidal rule, see equation (34), it can be concluded from equation (38) that the midpoint rule ($\theta = \frac{1}{2}$) is the only one which allows to compute the *exact* strain increment.

Remark 8. Because of equation (38), the second-order algorithm, equation (25), can be interpreted as a direct application of the midpoint rule.

Remark 9. Since equation (38) states that the midpoint rule is the only one which allows to compute the exact strain increment, the first-order algorithm, equation (23), is not deduced from the general trapezoidal rule particularized at $\theta = 0$. In an effort to obtain better accuracy, the first-order algorithm employs the *exact* strain increment expressed in a convenient form, equation (23b), instead of a forward-in-time *approximation* of $\Delta\varepsilon_{kl}$, such as $\Delta t\, {}^n d_{kl}$.

Remark 10. It must be observed that in Reference 17, Section 3.2, the midpoint rule only allows to compute the *most accurate* value of the strain increment, instead of the *exact* value. This behaviour of the midpoint rule was already pointed out by Key and Krieg.[19] The discrepancy is due to the different setting: material components are not used in those references.

A fixed frame is employed in Reference 17, and the strain increment is defined as

$$\Delta\varepsilon_{\alpha\beta} = \int_{t_n}^{t_{n+1}} d_{\alpha\beta}(t)\,\mathrm{d}t$$

However, according to Malvern[1] (Section 4.4) "*there is no physical significance to such an integration at one space point even in a steady flow simulation.*" On the other hand, Key and Krieg[19] work with the *co-rotational* components $D^{\alpha\beta}$ of the rate-of-deformation (which account for rigid rotations by means of the rotation tensor $\mathbf{R}$ coming from the polar decomposition of the deformation gradient), not with the Cartesian components $d_{\alpha\beta}$. In this context, they show that the midpoint rule yields a very accurate (but *not* exact) approximation to the logarithmic strain that results from the time-integration of $D^{\alpha\beta}$.

The midpoint rule also ensures that a second-order approximation for the position and the velocity is achieved,

$$^{n+1/2}z^\alpha = {}^{n+1/2}\bar{z}^\alpha + \mathcal{O}(\Delta t^2), \quad {}^{n+1/2}v^\alpha = \bar{v}^\alpha + \mathcal{O}(\Delta t^2) \tag{39}$$

By combining equations (35), (37) and (39), it is possible to conclude that second-order approximation is also obtained for the rate-of-deformation tensor,

$$^{n+1/2}d_{kl} = {}^{n+1/2}\bar{d}_{kl} + \mathcal{O}(\Delta t^2) \tag{40}$$

which, after substitution in equation (38) yields the relation employed for the error analysis, equation (28):

$$\Delta\varepsilon_{kl} = \Delta t\, {}^{n+1/2}d_{kl} + \mathcal{O}(\Delta t^3)$$

4. COMPUTATIONAL ASPECTS

4.1. From convected frames to Cartesian co-ordinates

The convected frame formalism may be employed to develop a large strain finite element code, thus resulting in a direct treatment of constitutive equations. Most existing large strain codes, however, use a fixed frame. Moreover, one of the goals of this paper is to present a simple way of enhancing a small strain code—written in a fixed frame—into a large strain code. For this reason, the adaptation of the algorithms presented above to a fixed orthonormal frame is shown next. It will be assumed that the initial orthonormal co-ordinates are employed as material co-ordinates, $x^i = Z^i$.

First-order algorithm

The transformation rule between the contravariant material and orthonormal components is shown in equation (13) for the stress tensor. A similar relation holds for the fourth-order modulus tensor, namely

$$C^{ijkl} = \sqrt{g}\frac{\partial Z^i}{\partial z^\alpha}\frac{\partial Z^j}{\partial z^\beta}\frac{\partial Z^k}{\partial z^\gamma}\frac{\partial Z^l}{\partial z^\delta}C^{\alpha\beta\gamma\delta} \tag{41}$$

Various representations of the strain increment in a fixed frame are possible, depending on which configuration they are referred to. Three common choices,[17] are the Lagrangian strain increment (referred to the configuration at t_n), the midstep strain increment (at $t_{n+1/2}$) and the Eulerian strain increment (at t_{n+1}). The first-order algorithm uses quantities at the beginning of the step, so the Lagrangian strain increment is chosen,

$$^n\Delta\varepsilon_{\mu\nu} = \frac{1}{2}\left[\frac{\partial(\Delta z^\mu)}{\partial^n z^\nu} + \frac{\partial(\Delta z^\nu)}{\partial^n z^\mu} + \frac{\partial(\Delta z^\alpha)}{\partial^n z^\nu}\frac{\partial(\Delta z^\alpha)}{\partial^n z^\mu}\right] \tag{42}$$

In this equation the superscript n emphasizes that $^n\Delta\varepsilon_{\mu\nu}$ are the orthonormal components of the strain increment referred to the configuration at t_n. By using the expression of the material components of the strain increment of the first-order algorithm, equation (23b), it can be checked that the relation between $\Delta\varepsilon_{kl}$ and $^n\Delta\varepsilon_{\mu\nu}$ is

$$\Delta\varepsilon_{kl} = \frac{\partial^n z^\mu}{\partial Z^k}\frac{\partial^n z^\nu}{\partial Z^l}{}^n\Delta\varepsilon_{\mu\nu} \tag{43}$$

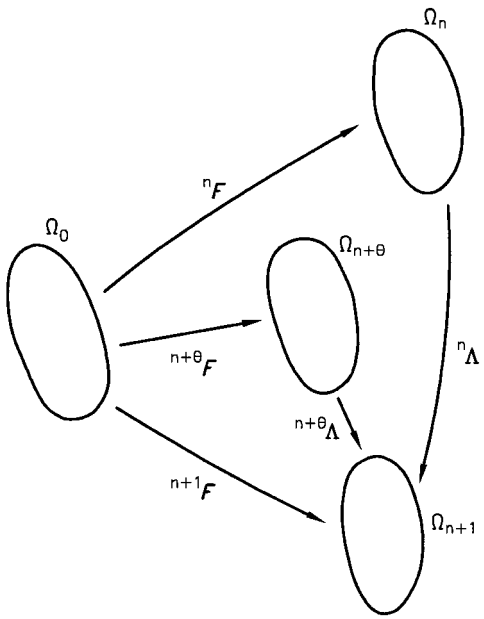


Figure 4. Incremental deformation gradients

Using the transformation rules given by equations (13), (41) and (43) in equation (23a) yields, after some manipulation,

$$^{n+1}\sigma^{\alpha\beta} = \frac{\sqrt{^n g}}{\sqrt{^{n+1}g}} \frac{\partial^{n+1} z^\alpha}{\partial Z^i} \frac{\partial^{n+1} z^\beta}{\partial Z^j} \frac{\partial Z^i}{\partial^n z^\gamma} \frac{\partial Z^j}{\partial^n z^\delta} \left(^n\sigma^{\gamma\delta} + {}^n C^{\gamma\delta\mu\nu} {}^n\Delta\varepsilon_{\mu\nu} \right) \tag{44}$$

The partial derivatives that appear in equation (44) are the orthonormal components of the deformation gradient $\mathbf{F}$ and its inverse, see equation (5) and Reference 1. In an incremental analysis, the incremental deformation gradient $^n\mathbf{\Lambda}$ that relates the configuration at the beginning of the time-step to the configuration at the end of the step,[3] see Figure 4, is defined as

$$^n\mathbf{\Lambda} = {}^{n+1}\mathbf{F} \cdot {}^n\mathbf{F}^{-1} \tag{45}$$

By modifying equation (44) according to equation (45) and taking into account that $\sqrt{g} = \det(\mathbf{F})$ (for the particular case of $x^i = Z^i$), the first-order algorithm is finally written in a fixed orthonormal frame, using compact notation, as

$$^{n+1}\boldsymbol{\sigma} = \frac{1}{\det(^n\mathbf{\Lambda})} {}^n\mathbf{\Lambda} \cdot [\, ^n\boldsymbol{\sigma} + {}^n\mathbf{C} : {}^n\Delta\boldsymbol{\varepsilon}] \cdot {}^n\mathbf{\Lambda}^{\mathrm{T}} \tag{46a}$$

$$^n\Delta\boldsymbol{\varepsilon} = \frac{1}{2} \left\{ \left[\frac{\partial(\Delta\mathbf{u})}{\partial^n\mathbf{z}} \right] + \left[\frac{\partial(\Delta\mathbf{u})}{\partial^n\mathbf{z}} \right]^{\mathrm{T}} + \left[\frac{\partial(\Delta\mathbf{u})}{\partial^n\mathbf{z}} \right]^{\mathrm{T}} \cdot \left[\frac{\partial(\Delta\mathbf{u})}{\partial^n\mathbf{z}} \right] \right\} \tag{46b}$$

where the superscript T means transpose.

Remark 11. The incremental deformation gradient ${}^{n}\mathbf{\Lambda}$ is employed in equation (46a) to push forward both the stress at time t_n and the stress increment, also referred to the configuration at t_n, to the final configuration at time t_{n+1}.

Remark 12. The midstep configuration is not needed in this algorithm. As a counterpart, however, quadratic terms are required to compute the strain increment referred to the configuration at t_n, equation (46b).

Second-order algorithm

A similar process is carried out for the second-order algorithm. Since this algorithm employs midstep quantities, the midstep strain increment,[17] is chosen to represent the increment of strain in the fixed frame,

$$n+1/2\overline{\Delta\varepsilon}_{\mu\nu} = \frac{1}{2}\left[\frac{\partial(\Delta z^{\mu})}{\partial^{n+1/2}\bar{z}^{\nu}} + \frac{\partial(\Delta z^{\nu})}{\partial^{n+1/2}\bar{z}^{\mu}}\right] \tag{47}$$

As previously commented, the bar indicates that ${}^{n+1/2}\overline{\Delta\varepsilon}_{\mu\nu}$ is referred to the midstep configuration associated to the constant-velocity assumption. To study the relation between $\Delta\varepsilon_{kl}$ and ${}^{n+1/2}\overline{\Delta\varepsilon}_{\mu\nu}$, it is convenient to use the expression of the material components of the strain increment of equation (25b). It can be checked then that

$$\Delta\varepsilon_{kl} = \frac{\partial^{n+1/2}\bar{z}^{\mu}}{\partial Z^{k}}\frac{\partial^{n+1/2}\bar{z}^{\nu}}{\partial Z^{l}}\,{}^{n+1/2}\overline{\Delta\varepsilon}_{\mu\nu} \tag{48}$$

Transforming equation (25a) from material to orthonormal components with the aid of the transformation rules of equations (13), (41) and (48) results in

$$\begin{aligned}{}^{n+1}\boldsymbol{\sigma}^{\alpha\beta} = {} & \frac{\sqrt{{}^{n}g}}{\sqrt{{}^{n+1}g}}\frac{\partial^{n+1}z^{\alpha}}{\partial Z^{i}}\frac{\partial^{n+1}z^{\beta}}{\partial Z^{j}}\frac{\partial Z^{i}}{\partial^{n}z^{\gamma}}\frac{\partial Z^{j}}{\partial^{n}z^{\delta}}\,{}^{n}\boldsymbol{\sigma}^{\gamma\delta} \\ & + \frac{\sqrt{{}^{n+1/2}g}}{\sqrt{{}^{n+1}g}}\frac{\partial^{n+1}z^{\alpha}}{\partial Z^{i}}\frac{\partial^{n+1}z^{\beta}}{\partial Z^{j}}\frac{\partial Z^{i}}{\partial^{n+1/2}\bar{z}^{\gamma}}\frac{\partial Z^{j}}{\partial^{n+1/2}\bar{z}^{\delta}}\,{}^{n+1/2}\overline{\mathbf{C}}^{\gamma\delta\mu\nu}\,{}^{n+1/2}\overline{\Delta\varepsilon}_{\mu\nu}\end{aligned} \tag{49}$$

Since the midstep configuration is involved, the incremental deformation gradient ${}^{n+1/2}\mathbf{\Lambda}$ is required, it is expressed as

$${}^{n+1/2}\mathbf{\Lambda} = {}^{n+1}\mathbf{F}\cdot{}^{n+1/2}\mathbf{F}^{-1} \tag{50}$$

and relates the midstep configuration to the end-of-step configuration, see Figure 4. With the help of ${}^{n+1/2}\mathbf{\Lambda}$, the second-order algorithm can be written in a fixed orthonormal frame as

$${}^{n+1}\boldsymbol{\sigma} = \frac{1}{\det({}^{n}\mathbf{\Lambda})}\,{}^{n}\mathbf{\Lambda}\cdot{}^{n}\boldsymbol{\sigma}\cdot{}^{n}\mathbf{\Lambda}^{\mathrm{T}} + \frac{1}{\det({}^{n+1/2}\mathbf{\Lambda})}\,{}^{n+1/2}\mathbf{\Lambda}\cdot[\,{}^{n+1/2}\bar{\boldsymbol{C}} : {}^{n+1/2}\overline{\Delta\boldsymbol{\varepsilon}}]\cdot{}^{n+1/2}\mathbf{\Lambda}^{\mathrm{T}} \tag{51a}$$

$${}^{n+1/2}\overline{\Delta\boldsymbol{\varepsilon}} = \frac{1}{2}\left\{\left[\frac{\partial(\Delta\mathbf{u})}{\partial^{n+1/2}\bar{\mathbf{z}}}\right] + \left[\frac{\partial(\Delta\mathbf{u})}{\partial^{n+1/2}\bar{\mathbf{z}}}\right]^{\mathrm{T}}\right\} \tag{51b}$$

Remark 13. The incremental deformation gradients ${}^{n}\mathbf{\Lambda}$ and ${}^{n+1/2}\mathbf{\Lambda}$ are employed in equation (51a) to push forward the stress at time t_n and the stress increment, referred to the midstep

configuration, to the final configuration at time t_{n+1}, so they can be properly added to yield a relevant tensor in this configuration. This transformation is essential, since orthonormal components of a tensor at different instants cannot be directly added, because they are referred to different configurations.[3]

Remark 14. The choice of the midstep configuration allows to compute the strain increment as the symmetrized gradient of the increment of displacements, with no quadratic terms. Because of this, equations (51) provide a straightforward generalization to large strains of the classical small strain analysis.

As commented previously the stress update algorithms present more elaborated expressions in a fixed frame, equations (46) and (51), than in a convected frame, equations (23) and (25). As explained in Remarks 11 and 13, quantities in a fixed frame must be transformed into a common configuration before they can be combined. If, from the start, a fixed frame context is chosen to state and time-integrate the rate-form constitutive equation, as done in References 2 and 3, these transformations are performed by means of pull-back and push-forward operators. A classical presentation of the two algorithms using an orthonormal fixed frame can be found in References 6 and 7.

4.2. Implementation in a small-strain code

It is shown in this subsection that both stress update algorithms can be employed to add large-strain capabilities to a small-strain FE code in a simple way.

The basic idea is that the incremental deformation gradients required in equations (46a) and (51a) can be computed in a straightforward manner using quantities that are available in a small strain code. Consider, for instance, the incremental deformation gradient ${}^{n}\mathbf{\Lambda}$ relating Ω_n to Ω_{n+1}, equation (45). Recalling the definition of $\mathbf{F}$ in equation (5) and the expression of incremental displacements, it can be easily checked that ${}^{n}\mathbf{\Lambda}$ can be written as

$$ {}^{n}\mathbf{\Lambda} = \frac{\partial^{n+1}\mathbf{z}}{\partial^{n}\mathbf{z}} = \mathbf{I} + \frac{\partial(\Delta\mathbf{u})}{\partial^{n}\mathbf{z}} \tag{52} $$

If an Updated Lagrangian formulation is used,[15] the configuration Ω_n is taken as a reference to compute the incremental displacements. In such a context, ${}^{n}\mathbf{\Lambda}$ can be computed from equation (52) with the aid of standard nodal shape functions, by expressing $\Delta\mathbf{u}$ in terms of the nodal values of incremental displacements. Since the derivatives of shape functions are available in a standard FE code,[12, 5] no new quantities must be computed to obtain ${}^{n}\mathbf{\Lambda}$.

As for ${}^{n+1/2}\mathbf{\Lambda}$, it can be computed as

$$ {}^{n+1/2}\mathbf{\Lambda} = \frac{\partial^{n+1}\mathbf{z}}{\partial^{n+1/2}\bar{\mathbf{z}}} = \mathbf{I} + \frac{1}{2}\frac{\partial(\Delta\boldsymbol{u})}{\partial^{n+1/2}\bar{\mathbf{z}}} \tag{53} $$

so ${}^{n+1/2}\mathbf{\Lambda}$ can also be directly computed with the aid of the shape functions, once the configuration of the mesh has been updated from Ω_n to $\Omega_{n+1/2}$.

As a result, the only two additional features that are required to handle large strains are (1) the updating of mesh configuration, and (2) the computation of incremental deformation gradients, equations (52) and (53). This can be seen by comparing the schematic algorithm for a small-strain analysis with non-linear material behaviour, shown in Box 1 with the large-strain versions, depicted in Box 2 (first stress update algorithm) and Box 3 (second stress update algorithm). In Boxes 2

Box 1. Small-strain analysis with non-linear material behaviour

For every time increment $[t_n, t_{n+1}]$;
For every iteration k within the time-increment;

1. Compute the incremental displacements $\Delta\mathbf{u}^k$ by solving a linearized form of the equilibrium equation
2. Compute the incremental strains $\Delta\boldsymbol{\varepsilon}^k$ as the symmetrized gradient of displacements:
$$\Delta\boldsymbol{\varepsilon}^k = \frac{1}{2}\left\{\left[\frac{\partial(\Delta\mathbf{u}^k)}{\partial\mathbf{Z}}\right] + \left[\frac{\partial(\Delta\mathbf{u}^k)}{\partial\mathbf{Z}}\right]^{\mathrm{T}}\right\}$$
3. Compute the elastic trial incremental stresses $\Delta\boldsymbol{\sigma}^k_{\mathrm{trial}}$ via the elastic modulus tensor:
$$\Delta\boldsymbol{\sigma}^k_{\mathrm{trial}} = \mathbf{C} : \Delta\boldsymbol{\varepsilon}^k$$
4. Compute the elastic trial stresses at t_{n+1}:
$$\boldsymbol{\sigma}^k_{\mathrm{trial}} = {}^n\boldsymbol{\sigma} + \Delta\boldsymbol{\sigma}^k_{\mathrm{trial}}$$
5. Compute the final stresses $\boldsymbol{\sigma}^k$ at t_{n+1} by performing the plastic correction
6. Compute the internal forces $\mathbf{f}^k_{\mathrm{int}}$ by integrating the stresses $\boldsymbol{\sigma}^k$
7. Check convergence. If it is not attained, go back to step 1

and 3 (large strains), the modifications with respect to Box 1 (small strains) are highlighted with boldface, and a star, $\star$, is employed to designate additional steps.

4.3. Computational cost of the algorithms

Regarding the computational cost of the stress update at each iteration, it can be seen in Boxes 2 and 3 that the second-order algorithm is more expensive. Indeed, it requires two configuration updates (from Ω_n to $\Omega_{n+1/2}$ and from $\Omega_{n+1/2}$ to Ω_{n+1}) and, more importantly, the computation of the incremental deformation gradients and Jacobians in two different configurations at each iteration. The first-order algorithm, on the other hand, only requires one computation for each iteration. This factor of two is an overestimation, because some steps of the stress update (such as the plastic correction) are performed once in both algorithms. However, it could be used as an approximation.

If the overall cost per iteration (and not only the stress update) is considered, other important and expensive features must be accounted for: evaluation of residuals, computation of stiffness matrices, resolution of the systems of equations, etc. And of course, this additional cost is basically independent of the particular stress update algorithm used. As a consequence, one can conclude that the overall cost per iteration is larger, but significantly less than twice, for the second stress update algorithm than for the first one.

Box 2. First stress update algorithm

For every time increment $[t_n, t_{n+1}]$;
For every iteration k within the time increment;

1. Compute the incremental displacements $\Delta\mathbf{u}^k$ by solving a linearized form of the equilibrium equation
⋆ **Compute the incremental deformation gradient** ${}^n\mathbf{\Lambda}$, **equation (52), and its determinant** ${}^nJ$
2. Compute the incremental strains $\Delta\boldsymbol{\varepsilon}^k$ **accounting for quadratic terms, equation (46b)**:

$$\Delta\boldsymbol{\varepsilon}^k = \frac{1}{2}\left\{\left[\frac{\partial(\Delta\mathbf{u}^k)}{\partial^n\mathbf{z}}\right] + \left[\frac{\partial(\Delta\mathbf{u}^k)}{\partial^n\mathbf{z}}\right]^{\mathrm{T}} + \left[\frac{\partial(\Delta\mathbf{u}^k)}{\partial^n\mathbf{z}}\right]^{\mathrm{T}} \cdot \left[\frac{\partial(\Delta\mathbf{u}^k)}{\partial^n\mathbf{z}}\right]\right\}$$

3. Compute the elastic trial incremental stresses $\Delta\boldsymbol{\sigma}^k_{\mathrm{trial}}$ via the elastic modulus tensor:

$$\Delta\boldsymbol{\sigma}^k_{\mathrm{trial}} = \mathbf{C} : \Delta\boldsymbol{\varepsilon}^k$$

4. Compute the elastic trial stresses at t_{n+1}, **pushing forward both** ${}^n\boldsymbol{\sigma}$ **and** $\Delta\boldsymbol{\sigma}^k_{\mathrm{trial}}$ **from configuration** Ω_n to Ω_{n+1}, **equation (46a)**:

$$\boldsymbol{\sigma}^k_{\mathrm{trial}} = {}^nJ^{-1}\ {}^n\mathbf{\Lambda}\ {}^n\boldsymbol{\sigma}\ {}^n\mathbf{\Lambda}^{\mathrm{T}} + {}^nJ^{-1}\ {}^n\mathbf{\Lambda}\left(\Delta\boldsymbol{\sigma}^k_{\mathrm{trial}}\right){}^n\mathbf{\Lambda}^{\mathrm{T}}$$

⋆ **Update the configuration from** Ω_n **to** Ω_{n+1} **by using the incremental displacements of the current step**
5. Compute the final stresses $\boldsymbol{\sigma}^k$ at t_{n+1} by performing the plastic correction
6. Compute the internal forces $\mathbf{f}^k_{\mathrm{int}}$ by integrating the stresses $\boldsymbol{\sigma}^k$
7. Check convergence. If it is not attained, **recover configuration** Ω_n and go back to step 1

Of course, the main interest is comparing the total computational cost for the full analysis, not the cost per iteration. In this context, the second algorithm clearly outperforms the first one, because the extra cost per iteration associated to the second algorithm is more than compensated by the decrease in the total number of iterations required. This behaviour is illustrated in this paper with a shell test.

This conclusion, however, cannot be readily extended to fast-transient dynamics, where explicit algorithms are typically employed for the time integration of the momentum balance. The use of explicit integration means that the overall computational cost is highly dependent on the cost of the stress update, because the additional features of an implicit algorithm (computing residuals and stiffness matrices; solving linear systems) are not required anymore. In this context, the cost of the stress update per time step (no iterations are performed) can become a decisive factor for preferring the first algorithm over the second one.

Box 3. Second stress update algorithm

For every time increment $[t_n, t_{n+1}]$;
For every iteration k within the time increment;

1. Compute the incremental displacements $\Delta\mathbf{u}^k$ by solving a linearized form of the equilibrium equation
- ⋆ **Compute the incremental deformation gradient ${}^{n}\mathbf{\Lambda}$, equation (52), and its determinant ${}^{n}J$**
- ⋆ **Update the configuration from Ω_n to $\Omega_{n+1/2}$**
2. Compute the incremental strains $\Delta\boldsymbol{\varepsilon}^k$ as the symmetrized gradient of displacements:

$$\Delta\boldsymbol{\varepsilon}^k = \frac{1}{2}\left\{\left[\frac{\partial(\Delta\mathbf{u}^k)}{\partial^{n+1/2}\bar{z}^\alpha}\right] + \left[\frac{\partial(\Delta\mathbf{u}^k)}{\partial^{n+1/2}\bar{z}^\alpha}\right]^{\mathrm{T}}\right\}$$

3. Compute the elastic trial incremental stresses $\Delta\boldsymbol{\sigma}^k_{\text{trial}}$ via the elastic modulus tensor:

$$\Delta\boldsymbol{\sigma}^k_{\text{trial}} = \mathbf{C} : \Delta\boldsymbol{\varepsilon}^k$$

- ⋆ **Compute the incremental deformation gradient ${}^{n+1/2}\mathbf{\Lambda}$, equation (53), and its determinant ${}^{n+1/2}J$**
4. Compute the elastic trial stresses at t_{n+1}, **pushing forward ${}^{n}\boldsymbol{\sigma}$ and $\Delta\boldsymbol{\sigma}^k_{\text{trial}}$ to Ω_{n+1}, equation (51a):**

$$\boldsymbol{\sigma}^k_{\text{trial}} = {}^{n}J^{-1}\ {}^{n}\mathbf{\Lambda}\ {}^{n}\boldsymbol{\sigma}\ {}^{n}\mathbf{\Lambda}^{\mathrm{T}} + {}^{n+1/2}J^{-1}\ {}^{n+1/2}\mathbf{\Lambda}\ \Delta\boldsymbol{\sigma}^k_{\text{trial}}\ {}^{n+1/2}\mathbf{\Lambda}^{\mathrm{T}}$$

- ⋆ **Update the configuration from $\Omega_{n+1/2}$ to Ω_{n+1}**
5. Compute the final stresses $\boldsymbol{\sigma}^k$ at t_{n+1} by performing the plastic correction
6. Compute the internal forces $\mathbf{f}^k_{\text{int}}$ by integrating the stresses $\boldsymbol{\sigma}^k$
7. Check convergence. If it is not attained, **recover configuration** Ω_n and go back to step 1

5. NUMERICAL EXAMPLES

5.1. Introduction

The two algorithms presented previously are compared with the help of various simple deformation paths (simple shear, uniaxial extension, extension and compression, extension and rotation) and two benchmark tests in non-linear mechanics: the necking of a circular bar and a shell under ring loads. Both elastic and elastoplastic cases are considered.

Elastic behaviour will be represented by a modulus tensor with constant orthonormal components which depend on the Lamé parameters λ and μ. All the simple deformation paths have been performed with a null Poisson's coefficient, resulting in $\lambda = 0$ and $\mu = E/2$, with E the Young's

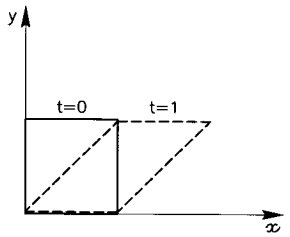


Figure 5. Simple shear test. Initial and final configurations

modulus. A bilinear elastoplastic law is employed for the plastic cases, with a plastic modulus of $E_p = E/100$ and a yield stress of $\sigma_y = E/2$.

A general result is that the values predicted with the first algorithm are more dependent on the number of time increments than for the second one. This behaviour, which is in agreement with the error analysis presented in Section 3, has been detected in the simple deformation paths and in both benchmark tests.

5.2. *Simple shear*

The problem statement for the simple shear test is shown in Figure 5, where the initial ($t = 0$) and final ($t = 1$) configurations are presented. The initial stress field is zero. The equations of motion are

$$\begin{aligned} x(t) &= X + Yt \\ y(t) &= Y \end{aligned} \tag{54}$$

If the rate-form hypoelastic behaviour is employed, the constitutive equation (in Cartesian co-ordinates) for this deformation path, equation (54), is the set of ordinary differential equations

$$\begin{aligned} \dot{\sigma}_{xx} - 2\sigma_{xy} &= 0 \\ \dot{\sigma}_{xy} - \sigma_{yy} &= E/2 \\ \dot{\sigma}_{yy} &= 0 \end{aligned} \tag{55}$$

which, complemented with null initial conditions, may be solved to provide the analytical solution

$$\begin{aligned} \sigma_{xx}(t) &= \frac{E}{2}t^2 \\ \sigma_{xy}(t) &= \frac{E}{2}t \\ \sigma_{yy}(t) &= 0 \end{aligned} \tag{56}$$

The two algorithms have been employed, with different values of the time step (1, 2, 3, 5, 10, 20, 50 increments), to integrate the constitutive equation for the deformation path (54), from $t = 0$ to $t = 1$. Figure 6 presents the results for the elastic case (1, 5, 50 increments). It can be seen that the second-order algorithm gets, for the three components of stress, the exact analytical values, whereas the first one grossly overestimates stresses when not enough increments are employed, and demands a small time step to get close to the analytical solution.

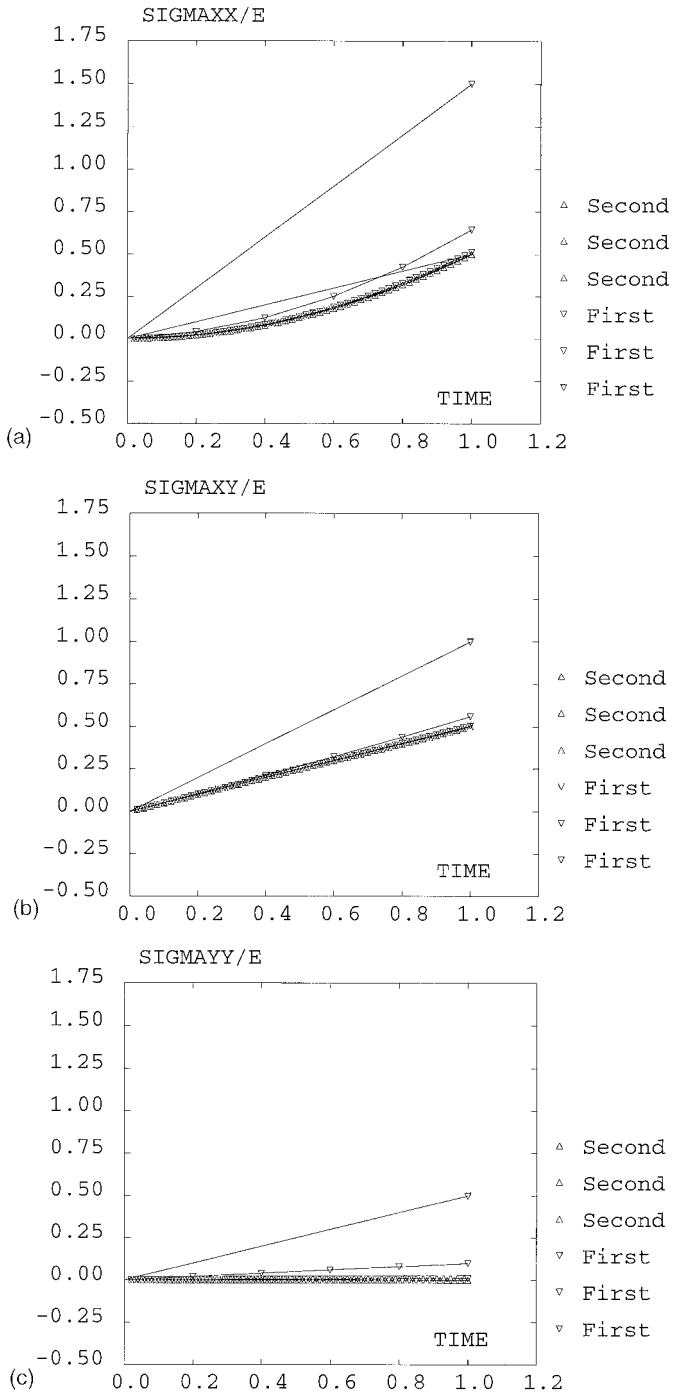


Figure 6. Simple shear test, elastic analysis. Stress vs. time curves computed with the two algorithms (1, 5 and 50 time steps): (a) σ_{xx}, (b) σ_{xy}, (c) σ_{yy}

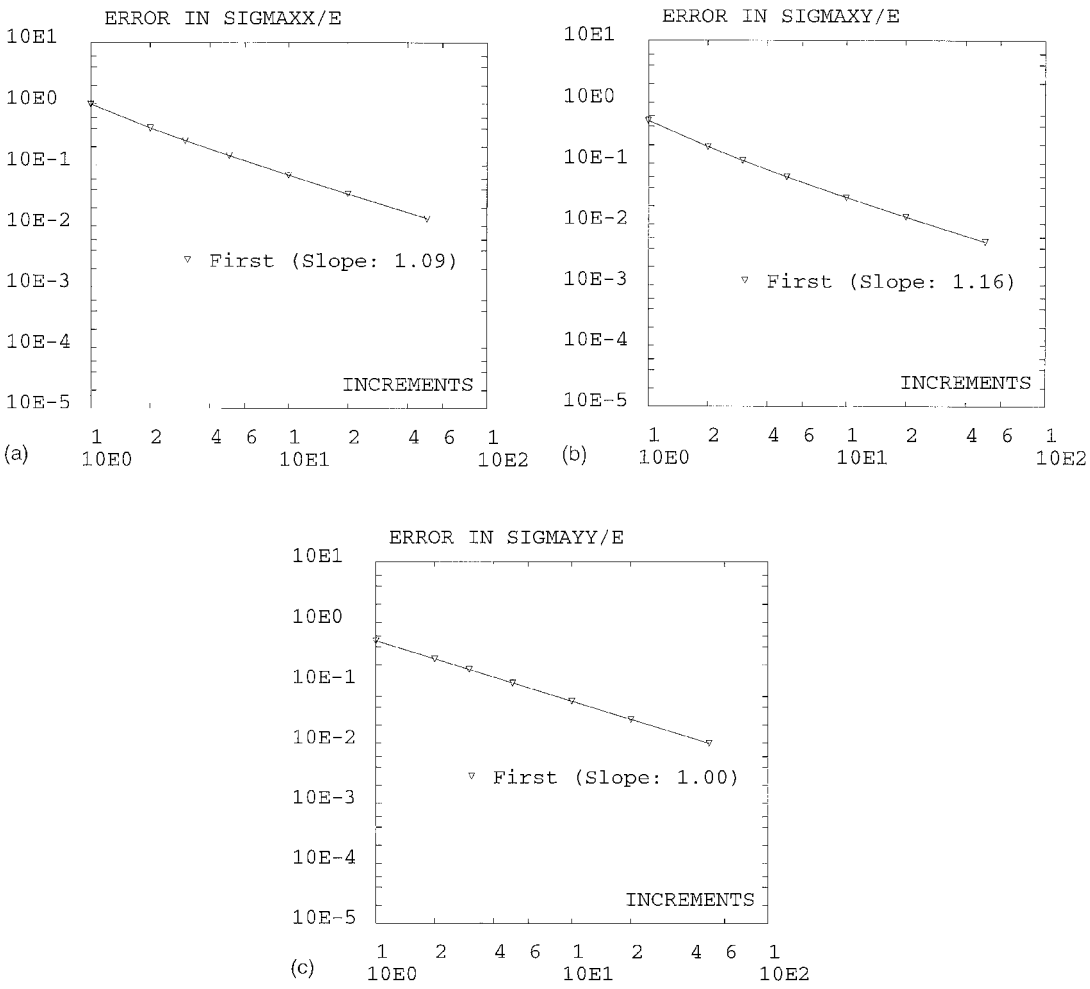


Figure 7. Simple shear test, elastic analysis. Error in the final value of stress ($t = 1$) vs. number of time steps (1, 2, 3, 5, 10, 20, 50): (a) σ_{xx}, (b) σ_{xy}, (c) σ_{yy}

Taking the analytical solution (56), the error in the final stress ($t = 1$) can be computed and plotted versus the number of time increments. In a log–log scale, a straight line with a slope equal to the order of the algorithm is expected (i.e., 1 for the first algorithm and 2 for the second). To compute an average observed slope, a straight line is fitted via least-squares interpolation. The results are shown in Figure 7. Only the first algorithm is shown, because the second one provides the exact analytical solution (except for rounding errors) for any number of time steps. It can be seen that for the three components of stress, Figures 7(a), 7(b) and 7(c), the observed slopes are very close to the expected value of 1 (1·09 for σ_{xx}, 1·16 for σ_{xy} and 1·00 for σ_{yy}). The first-order accuracy of the first algorithm is thus corroborated by this simple test.

The shear test has also been performed in the elastoplastic case. The results are presented in Figure 8 (1, 10, 50 increments). As in the elastic problem, the first-order algorithm grossly

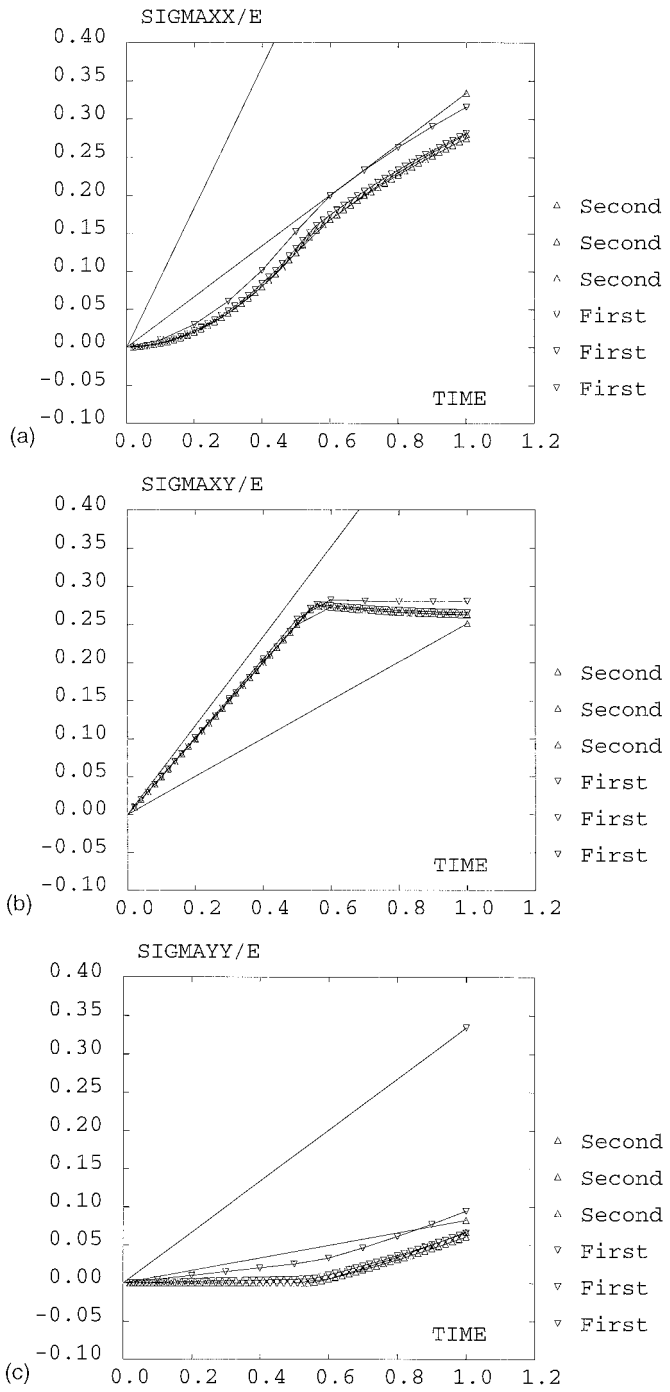


Figure 8. Simple shear test, elastoplastic analysis. Stress vs. time curves computed with the two algorithms (1, 10 and 50 time steps): (a) σ_{xx}, (b) σ_{xy}, (c) σ_{yy}

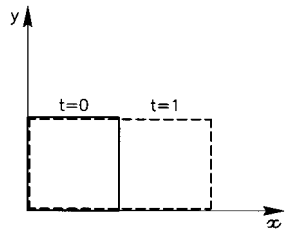


Figure 9. Uniaxial extension test. Initial and final configurations

overestimates the final stress if only one increment is employed, while the second-order one provides much more accurate values. With a higher number of time steps, both algorithms converge to the same response. It may be observed that, of course, when plastification starts (around $t = 0.6$), the curves differ from their elastic counterparts of Figure 6.

5.3. Uniaxial extension

A unit square is subjected to uniaxial extension in the x direction, see Figure 9. The equations of motion are

$$\begin{aligned} x(t) &= X(1+t) \\ y(t) &= Y \end{aligned} \tag{57}$$

and the analytical solution of the hypoelastic constitutive equation is

$$\begin{aligned} \sigma_{xx}(t) &= Et \\ \sigma_{xy}(t) &= 0 \\ \sigma_{yy}(t) &= 0 \end{aligned} \tag{58}$$

The two algorithms correctly predict null values for σ_{xy} and σ_{yy}. Differences are found, on the contrary, for σ_{xx}: while the first algorithm grossly overestimates it, the second one slightly underestimates it, see Figure 10. The error in the final value of σ_{xx} versus the number of time steps is presented in Figure 11. The observed slopes for the two algorithms are in this case 1·13 for the first algorithm and 1.95 for the second one, close to expected values of 1 and 2, respectively.

The plastic response is presented in Figure 12. When plastification begins at $t = 0{\cdot}5$, the plastic correction is performed along the deviatoric part of the stress tensor, thus resulting in an increase of σ_{yy} at the expense of σ_{xx}. Again, the second-order algorithm behaves much better than the first one if a small number of time-increments is employed, especially for σ_{xx}, see Figure 12(a).

5.4. Extension and compression

A unit square undergoes extension in the x direction and compression in the y direction, with no change in volume, see Figure 13. The equations of motion are

$$\begin{aligned} x(t) &= X(1+t) \\ y(t) &= Y/(1+t) \end{aligned} \tag{59}$$

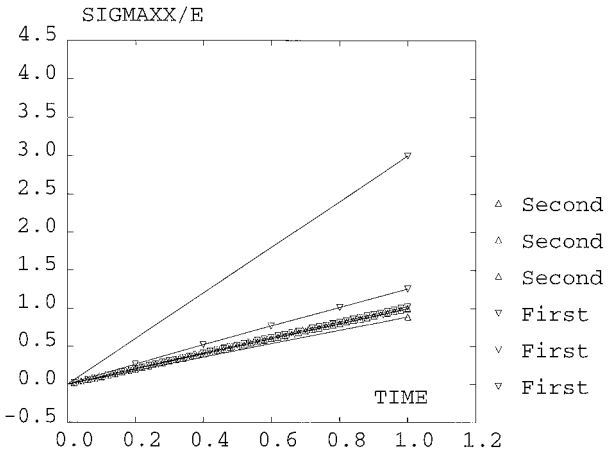


Figure 10. Uniaxial extension test, elastic analysis. Horizontal normal stress σ_{xx} vs. time curves computed with the two algorithms (1, 5 and 50 time steps)

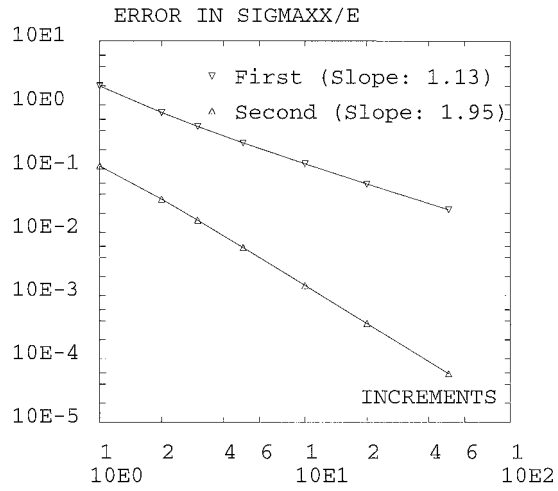


Figure 11. Uniaxial extension test, elastic analysis. Error in the final value of horizontal normal stress σ_{xx} $(t=1)$ vs. number of time steps (1, 2, 3, 5, 10, 20, 50)

and the analytical solution for the elastic case is

$$\begin{aligned} \sigma_{xx}(t) &= E\left(t+\frac{t^2}{2}\right) \\ \sigma_{xy}(t) &= 0 \\ \sigma_{yy}(t) &= \frac{E}{2}\left[\frac{1}{(1+t)^2}-1\right] \end{aligned} \tag{60}$$

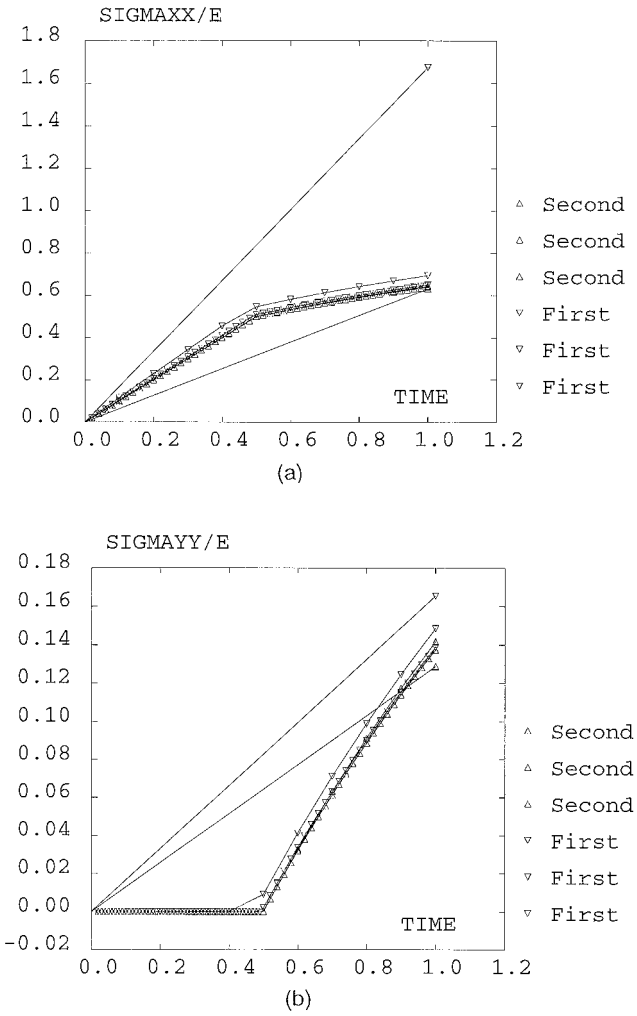


Figure 12. Uniaxial extension test, elastoplastic analysis. Stress vs. time curves computed with the two algorithms (1, 10 and 50 time steps): (a) σ_{xx}, (b) σ_{yy}

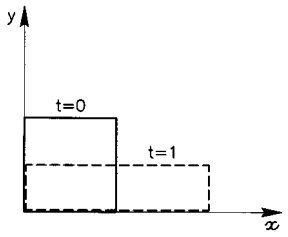


Figure 13. Extension and compression test. Initial and final configurations

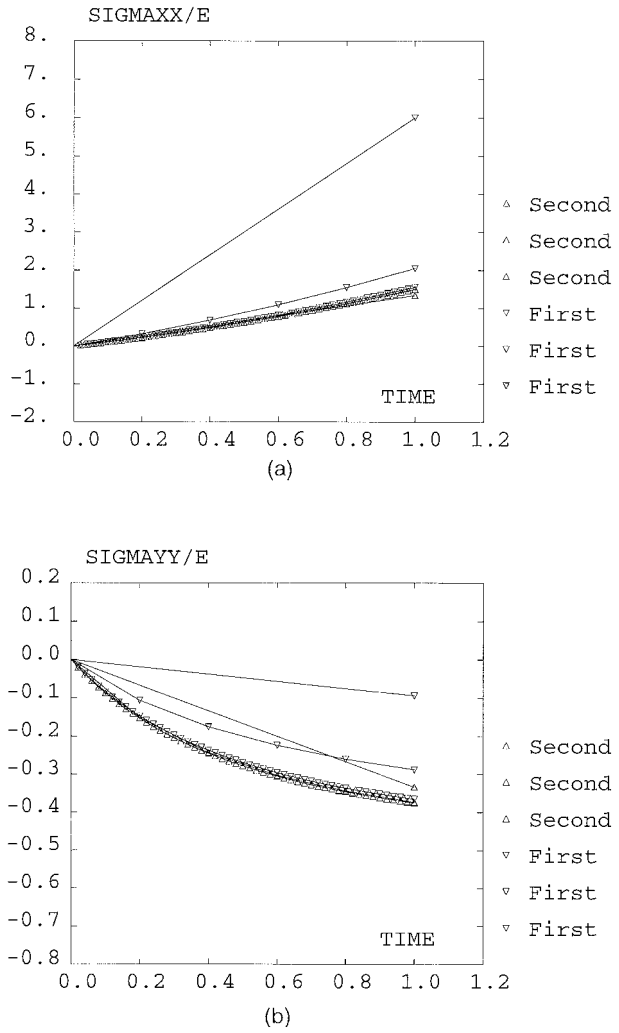
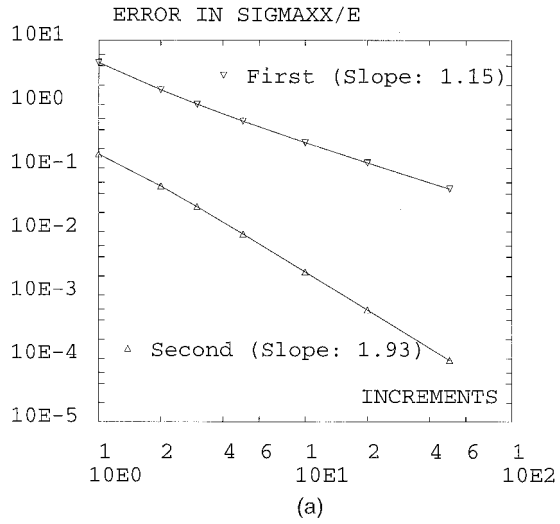


Figure 14. Extension and compression test, elastic analysis. Stress vs. time curves computed with the two algorithms (1, 5 and 50 time steps): (a) σ_{xx}, (b) σ_{yy}

Again, both algorithms are capable of predicting null σ_{xy} for the elastic test, but differences appear for σ_{xx} and σ_{yy}, see Figure 14. The observed orders are in this case 1·15 (first algorithm) and 1·93 (second algorithm) for σ_{xx}, see Figure 15(a), and 0·87 (first algorithm) and 1·99 (second algorithm) for σ_{yy}, see Figure 15(b).

As for the plastic test, results are shown in Figure 16. Once again, convergence to the "reference" solution (with 50 time increments) is faster with the second algorithm.



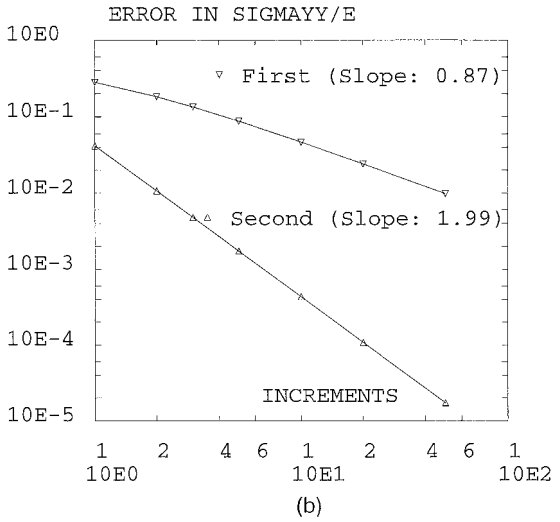


Figure 15. Extension and compression test, elastic analysis. Error in the final value of stress ($t=1$) vs. number of time steps (1, 2, 3, 5, 10, 20, 50): (a) σ_{xx}, (b) σ_{yy}

5.5. *Extension and rotation*

In this last deformation path, a unit square undergoes a uniaxial extension and a superposed rigid rotation, see Figure 17. The equations of motion are

$$\begin{aligned} x(t) &= X(1+t)\cos(2\pi t) - Y\sin(2\pi t) \\ y(t) &= X(1+t)\sin(2\pi t) + Y\cos(2\pi t) \end{aligned} \tag{61}$$

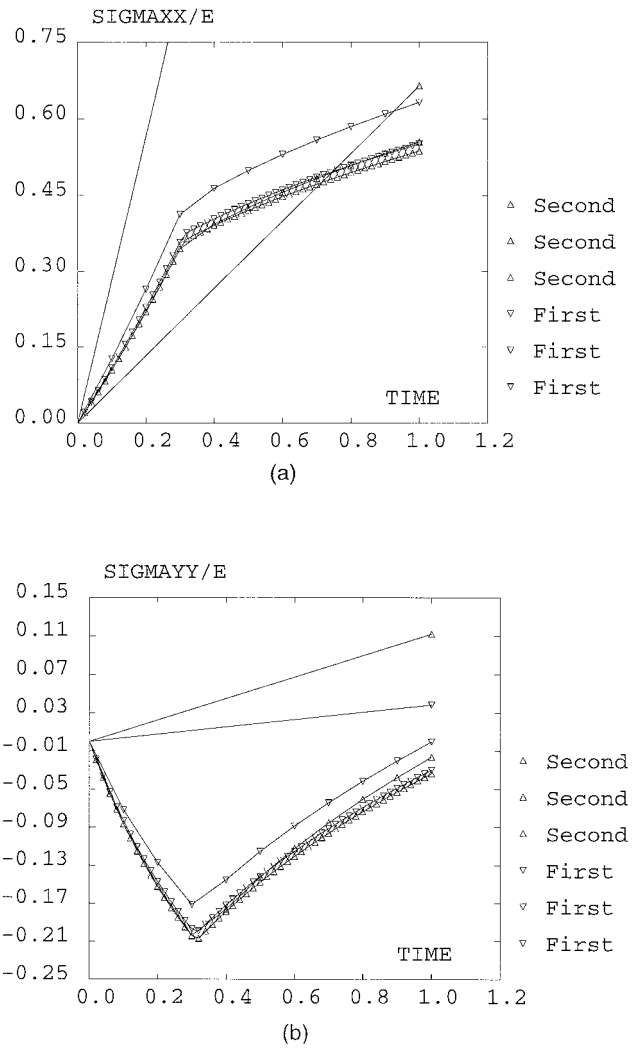


Figure 16. Extension and compression test, elastoplastic analysis. Stress vs. time curves computed with the two algorithms (1, 10 and 50 time steps): (a) σ_{xx}, (b) σ_{yy}

and the analytical solution is

$$\begin{aligned} \sigma_{xx}(t) &= Et\cos^2(2\pi t) \\ \sigma_{xy}(t) &= Et\sin^2(2\pi t) \\ \sigma_{yy}(t) &= Et\sin(2\pi t)\cos(2\pi t) \end{aligned} \tag{62}$$

which is, as expected, a rotation of the solution to the uniaxial extension test, equation (58).

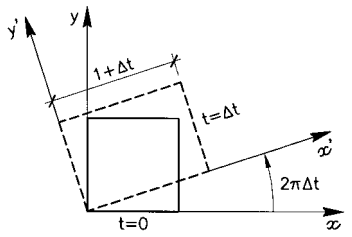


Figure 17. Extension and rotation test. Problem statement

The output of the elastic analysis with 1, 5 and 50 time steps can be seen in Figure 18. If only one increment is employed, the rotation part of the motion is not captured and the predicted stress is identical to that of the uniaxial extension test. With a higher number of steps, the comparative performance of the two algorithms is different from that of the previous tests. For the stress components σ_{xy} and σ_{yy}, for instance, the first algorithm is the one which predicts correct null final values for any number of time increments. In fact, this result illustrates a more general behaviour. As shown in Reference 7, for a general shear-free deformation path (i.e. the original square is transformed into a rectangle), the first algorithm correctly predicts null shear stresses for any number of time steps, while the second one does not.

As for the stress component σ_{xx}, the first algorithm performs better if a reduced number of time steps (5) is employed, and a larger number is required for the second algorithm to produce more accurate results. This behaviour is illustrated by Figure 19(a), where the error curves for σ_{xx} of the two algorithms intersect each other. Again, the observed order of both schemes (1·07 for the first one and 2·30 for the second one) is in accordance with the expected values. Figures 19(b) and 19(c) show the convergence behaviour of the second algorithm for the other two stress components. It can be seen that the convergence to the exact analytical value is very fast, especially for σ_{yy}, Figure 19(c). The outcome of the plastic analysis is depicted in Figure 20.

5.6. Necking of a circular bar

The necking problem is a well-known benchmark test in non-linear solid mechanics.[20] A circular bar, with a radius of 6·413 and 53·334 mm length, is subjected to uniaxial tension. A slight geometric imperfection (1 per cent radius reduction) induces necking in the central part of the bar. Because of axisymmetry, only a quarter of the specimen is modelled. The uniaxial constitutive law may be found in Reference 20.

A mesh of 50 eight-noded quadrilaterals with 2×2 Gauss points is employed to simulate an axial elongation of 14 mm (26 per cent of initial length). To assess the time increment sensitivity of each algorithm, the computation is performed with three different time increments (100, 200, and 1000 steps) for both algorithms.

The final deformed shape of the specimen is shown, for the reference solution (second algorithm with 1000 time steps), in Figure 21. The influence of the time-increment can be seen in Figure 22, where radius reduction is plotted versus elongation. The differences are important for the first algorithm, see Figure 22(a), starting at around 10 per cent elongation and leading to significantly different final values of the neck radius (21 per cent). On the other hand, the second

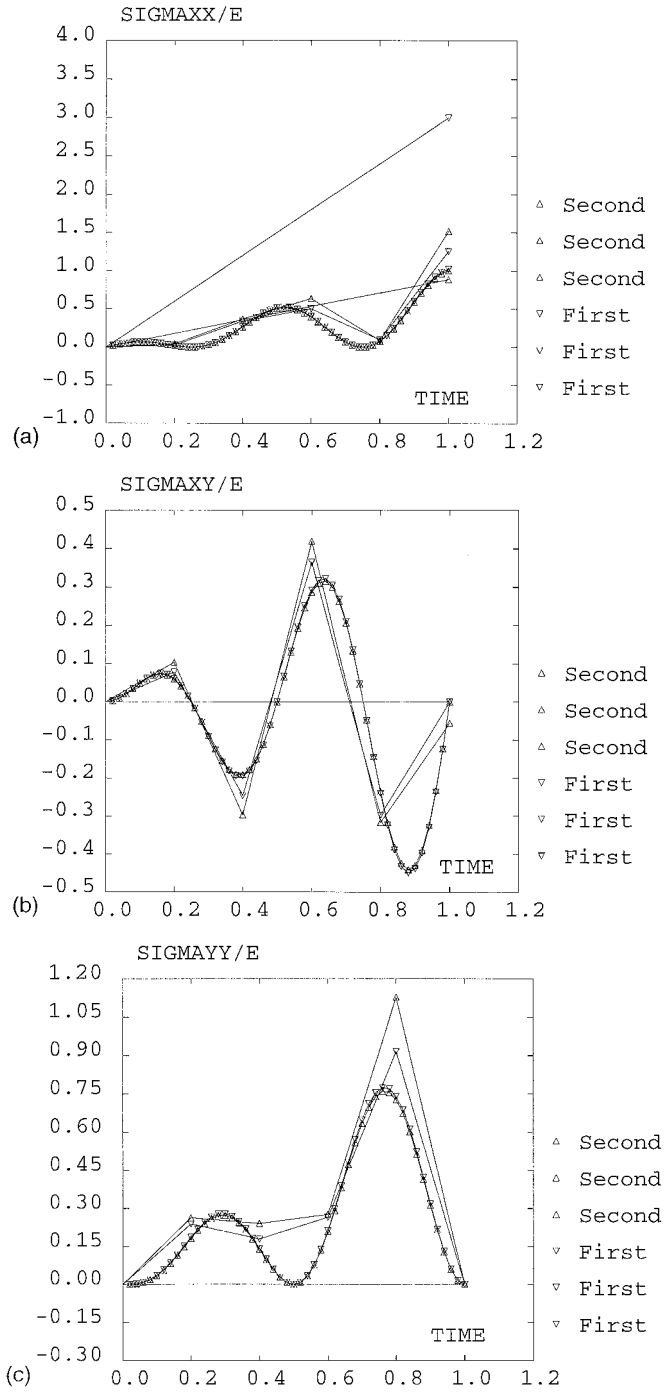


Figure 18. Extension and rotation test, elastic analysis. Stress vs. time curves computed with the two algorithms (1, 5 and 50 time steps): (a) σ_{xx}, (b) σ_{xy}, (c) σ_{yy}

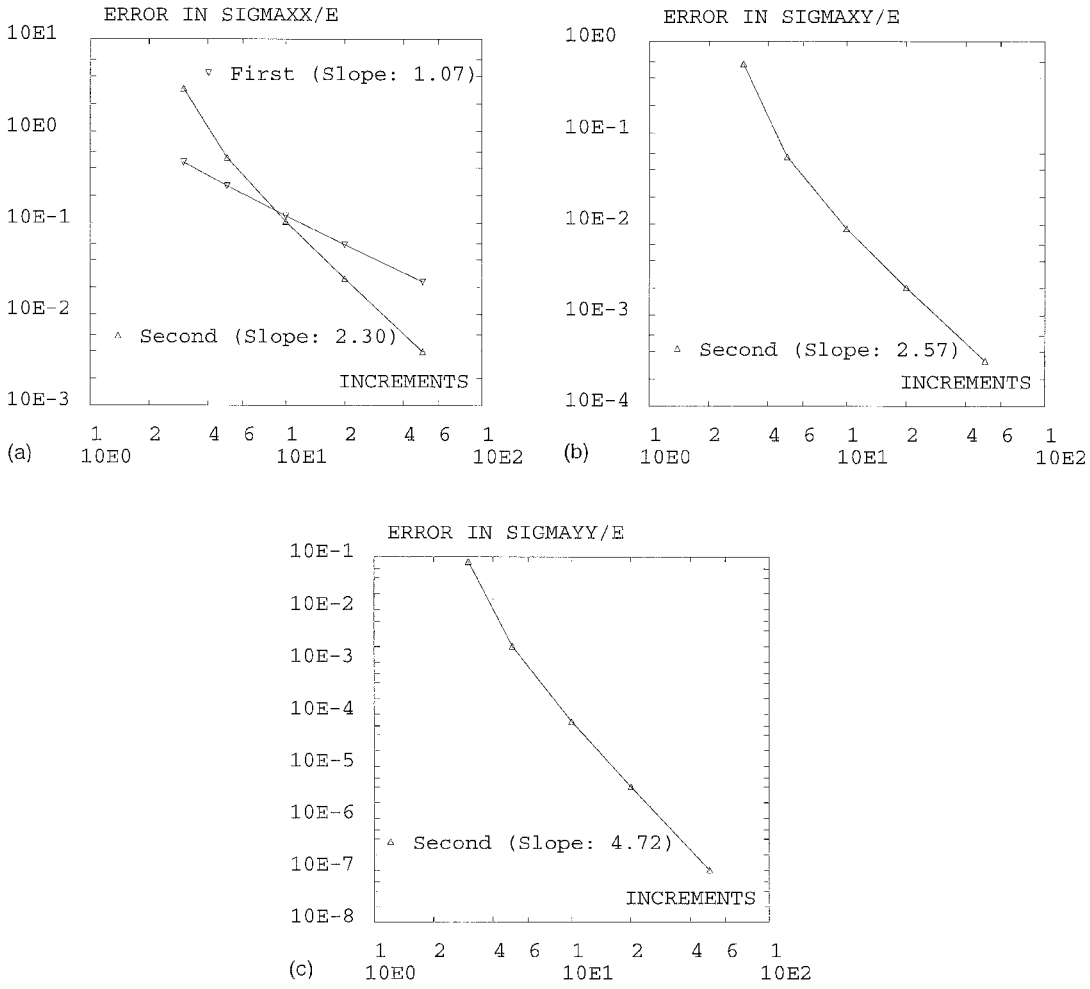


Figure 19. Extension and rotation test, elastic analysis. Error in the final value of stress ($t = 1$) vs. number of time steps (3, 5, 10, 20, 50): (a) σ_{xx}, (b) σ_{xy}, (c) σ_{yy}

algorithm shows much less sensitivity to the time increment: the three curves, see Figure 22(b), are much closer together, with the 200 and 1000 time step solutions almost superimposed and with a discrepancy in the final neck radius of only 3 per cent. Moreover, these last curves are very similar to those portrayed in Reference 20.

5.7. Shell test

An axisymmetric shell made of an elastic material is clamped at its border, and a ring load P is applied to the shell, causing a deflection v of the apex, see Figure 23. This is a classical benchmark test in geometrically non-linear structural analysis,[21, 5] and will be employed here to assess the performance of the two stress update algorithms for structural finite elements.

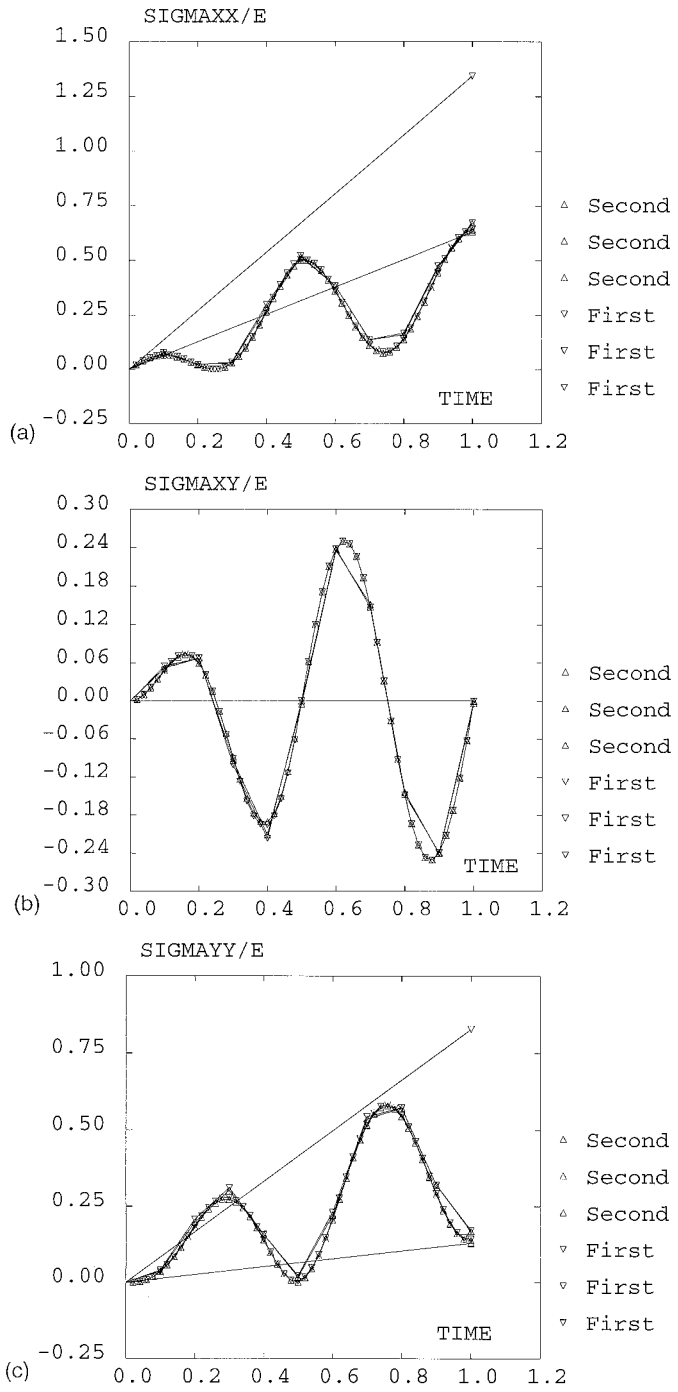


Figure 20. Extension and rotation test, elastoplastic analysis. Stress vs. time curves computed with the two algorithms (1,10 and 50 time steps): (a) σ_{xx}, (b) σ_{xy}, (c) σ_{yy}

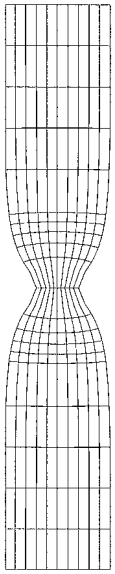


Figure 21. Necking test. Final deformed shape. Second algorithm with 1000 time steps

This problem is solved in Reference 5 with a displacement-controlled technique,[16] on the apex (i.e. by prescribing increasing values of v), and for different values of the load eccentricity $e = r/R_h$. For comparison purposes, the maximum value of $e = 0{\cdot}42$ employed in Reference 5, has been chosen. The analysis has been performed with two values of the increment of v: $\Delta v = 0{\cdot}005$ in and $\Delta v = 0{\cdot}001$ in.

Figure 24 shows the load–deflection curves for the two stress update algorithms. It can be seen that the response for the first algorithm, Figure 24(a), is much more dependent on the size of Δv than for the second algorithm, Figure 24(b). This shows again the superior accuracy of the second algorithm. It must be remarked that the solution with the first algorithm and $\Delta v = 0{\cdot}001$ in converges to the solution with the second algorithm which, moreover, shows good agreement with that of Reference 5 (with a load peak around $P = 75$ lb).

For larger values of the eccentricity e, a displacement-controlled method is no longer valid, because the load–deflection curve shows a snap-back behaviour. An arc-length technique,[16] is then necessary, see Reference 21. A cylindrical arc-length formulation, combined with the full Newton–Raphson method, has been used to solve the problem with $e = 0 \cdot 60$. The arc length is automatically updated at every step as proposed in Reference 16, by prescribing a desired number of four iterations per step. An initial arc length of $\Delta \ell = 0{\cdot}01$ in and a lower bound of $\Delta \ell = 0{\cdot}005$ in have been employed.

The load–deflection curves are presented in Figure 25. The computational cost for the two algorithms is shown in Table I. It can be seen that the required number of both time steps and iterations for the first algorithm is about twice as much as for the second algorithm. As already remarked in Section 4.3, this results in a higher computational efficiency of the second-order algorithm.

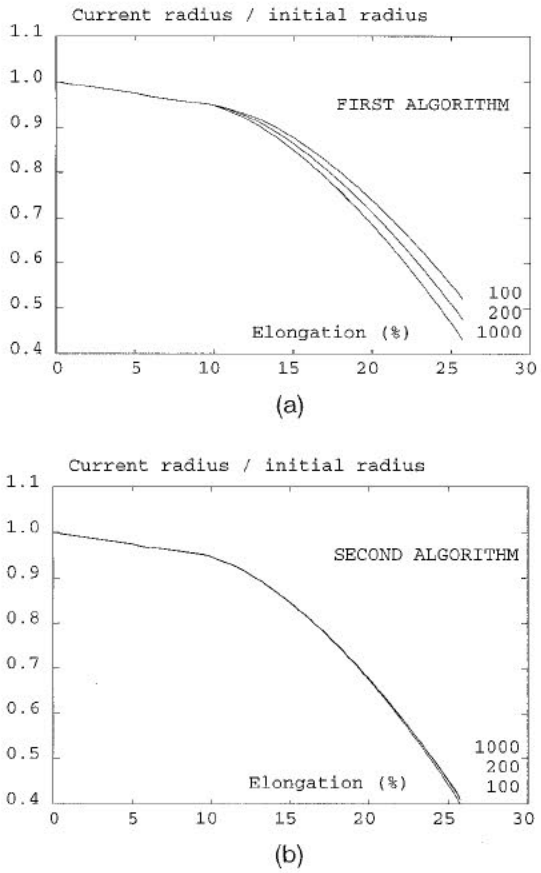


Figure 22. Radius reduction vs. bar elongation curves computed with 100, 200 and 1000 time steps: (a) first algorithm, (b) second algorithm

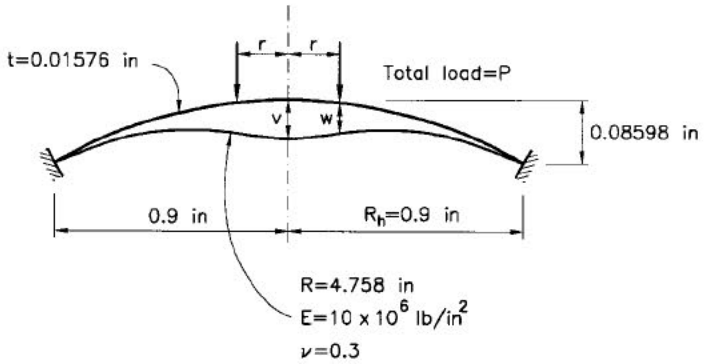


Figure 23. Shell test. Problem statement

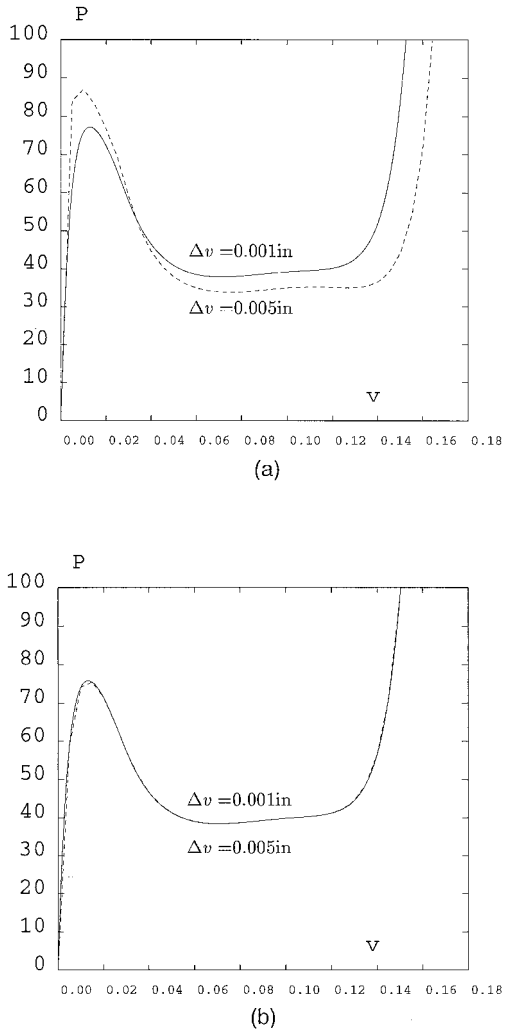


Figure 24. Load vs. deflection curves for an eccentricity $e = 0{\cdot}42$. Displacement-control solutions with $\Delta v = 0{\cdot}005$ in and $\Delta v = 0{\cdot}001$ in: (a) first algorithm, (b) second algorithm

It is interesting to note that the average number of iterations per step is 4·1 for the first algorithm and 4·2 for the second one, thus indicating a very good performance of the automatic arc-length control.[16] Because of its second-order accuracy, the second algorithm shows superior global convergence properties, thus allowing for larger arc lengths. This point is clear from Figure 26, which depicts the arc length versus the accumulated length ℓ. Except for a few steps, the first algorithm generally demands smaller $\Delta\ell$ to satisfy the requirement of four iterations per step. As a result, the total number of iterations is higher for the first algorithm. Figure 27 shows the accumulated number of iterations versus the accumulated length ℓ.

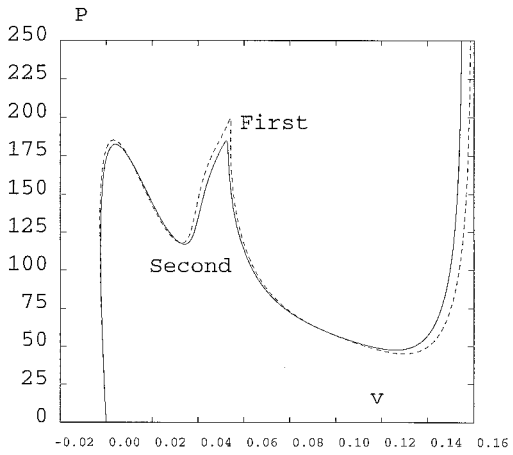


Figure 25. Load vs. deflection curves for an eccentricity $e = 0{\cdot}60$. Arc-length solutions with automatic arc-length control

Table I. Computational cost for the shell test

	Load steps	Iterations	Iterations per step
First algorithm	418	1719	4·1
Second algorithm	198	822	4·2

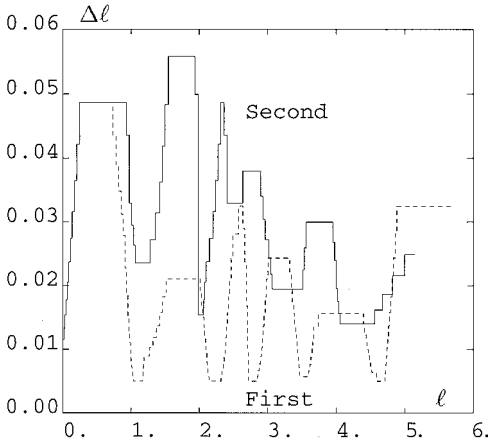


Figure 26. Arc length vs. accumulated length

As a final remark, it is worth mentioning that the second algorithm yields a more accurate load–deflection response in Figure 25. This fact can be verified by reproducing the test with the first algorithm and a very small constant $\Delta\ell = 0.005$ in (i.e. the lower bound previously used, with no automatic update of the arc length). A total of 1098 steps and 3643 iterations are required.

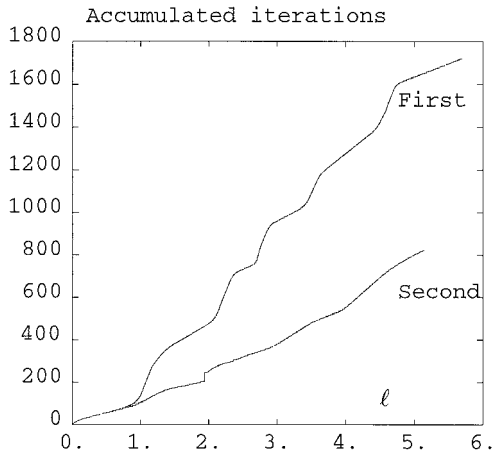


Figure 27. Accumulated iterations vs. accumulated length

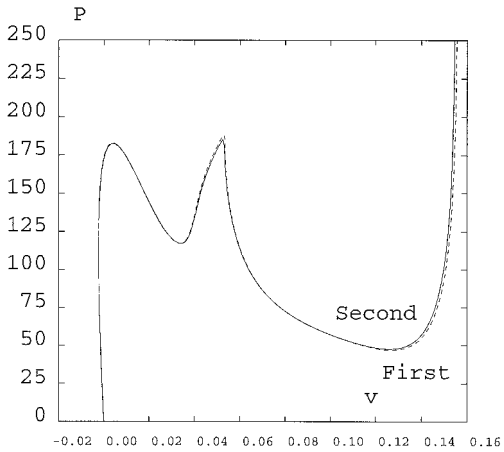


Figure 28. Load vs. deflection curves. Constant $\Delta\ell = 0{\cdot}005$ in for the first algorithm

As shown in Figure 28, the solution then converges to that obtained with the second algorithm with only 198 steps and 822 iterations. In conclusion, the second algorithm provides a clearly better solution, both in terms of accuracy and computational cost.

6. CONCLUDING REMARKS

A methodology to compare stress update algorithms for large strains from an analytical point of view has been presented and applied to two simple algorithms for hypoelastic constitutive laws.

An accuracy analysis has been performed to deduce the order of the truncation error associated to each numerical algorithm. This provides an *a priori* knowledge on the accuracy for each algorithm. Therefore, the performance of the algorithms can be studied from an analytical point of view, in addition to the usual numerical experiments.

The basic ingredient of this approach is the use of convected frames. This choice enables standard techniques of numerical analysis in finite differences to evaluate the accuracy of the numerical time integration.

The convected frame formalism also allows a unified derivation of both algorithms. The first algorithm,[2] which is first-order-accurate, uses the full incremental Lagrange strain tensor as the strain measure. The second one measures strain with the usual small strain tensor, but computed in the midstep configuration. This algorithm,[3] is second-order-accurate.

Moreover, it has been shown that the main difference, from an accuracy point of view, resides in the elastic modulus tensor (in particular when it is evaluated). Because both algorithms use the *exact* value for the strain increment.

After the accuracy analysis, the two stress update algorithms are adapted to a fixed frame setting. In this manner, they can be employed to add large strain capabilities to an existing finite element code for non-linear analysis. Various numerical tests are then performed to validate the implementation and to compare the algorithms. These tests corroborate the *a-priori* accuracy analysis presented here.

Regarding the computational efficiency, the second algorithm is superior for the quasistatic problems considered here, where an implicit integration is performed. This conclusion, however, cannot be readily extended to explicit algorithms for fast-transient dynamics.

A set of simple deformation paths (simple shear, uniaxial extension, extension and compression, extension and rotation) have been used to assess the relative performance of the two algorithms, both for large-strain elastic and elastoplastic analysis. The general outcome of these numerical tests is in good agreement with the accuracy analysis: the predicted solutions are much more dependent on the time step for the first-order algorithm than for the second-order one. The extension and rotation test, however, illustrates that the first-order algorithm can have a superior performance for certain stress components in some deformation paths. Finally, two well-documented benchmark tests, a necking analysis and a shell under a ring load, confirm the previous conclusions and show the superior performance of the second-order algorithm with continuum and structural elements.

APPENDIX

The midstep material components of the modulus tensor

It is shown in this appendix that the constant-velocity assumption yields a second-order approximation to the exact midstep material components of the modulus tensor. This property, equation (32), is essential for the second-order accuracy of the second stress update algorithm.

It is assumed that the modulus tensor $\mathbf{C}$ depends on the deformation,[3] represented in a material setting by the metric tensor $\mathbf{G}$. That is, the dependence of $\mathbf{C}$ with respect to time is not explicit, but associated to the deformation of the body, and can be written as

$$C^{ijkl}(t) = f(g_{mn}(t)) \tag{63}$$

where f is a function of the components of the metric tensor g_{mn}. The exact midstep components of the modulus tensor are then

$$^{n+1/2}C^{ijkl} = f(^{n+1/2}g_{mn}) \tag{64}$$

where $^{n+1/2}g_{mn}$ are the exact midstep components of the metric tensor, while the approximate midstep components of $\mathbf{C}$ are

$$^{n+1/2}\overline{C}^{ijkl} = f(^{n+1/2}\bar{g}_{mn}) \tag{65}$$

with $\bar{g}_{mn}$ the constant-velocity approximation of g_{mn}. Recalling the definition of g_{mn}, equation (8), and the relation between exact and approximate midstep orthonormal co-ordinates, $^{n+1/2}z^{\alpha}$ and $^{n+1/2}\bar{z}^{\alpha}$, equation (39), it can be easily concluded that

$$^{n+1/2}g_{mn} = {}^{n+1/2}\bar{g}_{mn} + \mathcal{O}(\Delta t^2) \tag{66}$$

As previously shown for other quantities, the constant-velocity assumption also provides a second-order approximation to the components of the metric tensor. Substituting equation (66) into equation (64) and performing a first-order Taylor's expansion finally renders, assuming that the first derivative of f is bounded,

$$^{n+1/2}C^{ijkl} = {}^{n+1/2}\overline{C}^{ijkl} + \mathcal{O}(\Delta t^2)$$

which is equation (32).

REFERENCES

1. L. W. Malvern, *Introduction to the Mechanics of a Continuous Medium*, Prentice-Hall Series in Engineering of the Physical Sciences, Prentice-Hall, Englewood Cliffs, NJ, 1969.
2. K. J. Bathe, E. Ramm and E. L. Wilson, 'Finite element formulations for large deformation dynamic analysis', *Int. J. Numer. Meth. Engng.*, **9,** 353–386 (1975).
3. P. M. Pinsky, M. Ortiz and K. S. Pister, 'Numerical integration of rate constitutive equations in finite deformation analysis', *Comput. Methods Appl. Mech. Engng.*, **40,** 137–158 (1983).
4. P. Pegon and P. Guélin, 'Finite strain plasticity in convected frames', *Int. J. Numer. Meth. Engng.*, **22,** 521–545 (1986).
5. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, Vols. 1 and 2, McGraw Hill, London, 1991.
6. A. Rodríguez-Ferran and A. Huerta, 'A comparison of two objective stress rates in object-oriented codes', Monograph No. 26, International Centre for Numerical Methods in Engineering, Barcelona. ISBN: 84-87867-52-9, 1994.
7. A. Rodríguez-Ferran and A. Huerta, 'Comparing two algorithms to add large strains to a small strain finite element code', Research Report no. 91, International Centre for Numerical Methods in Engineering, Barcelona, 1996.
8. C. Truesdell and R. Toupin, in: S. Flügge (ed.), *The Classical Field Theories*, *Encyclopedia of Physics*, Vol. III/1, Springer, Berlin, 1960.
9. J. G. Oldroyd, 'On the formulation of rheological equations of state', *Proc. Roy. Soc. London*, **A200,** 523–541 (1950).
10. C. Truesdell and W. Noll, in: S. Flügge (ed.), *The Non-Linear Field Theories of Mechanics*, *Encyclopedia of Physics*, Vol. III/3, Springer, Berlin, 1965.
11. L. Brillouin, *Les tenseurs en mécanique et en élasticité*, Masson, Paris, 1949.
12. T. J. R. Hughes, *The Finite Element Method*, Prentice-Hall, Stanford, 1987.
13. C. Truesdell, 'Corrections and additions to 'The mechanical foundations of elasticity and fluid dynamics'', *J. Rat. Mech. Anal.*, **2,** 505–616 (1953).
14. C. Truesdell, 'The simplest rate theory of pure elasticity', *Commun. Pure Appl. Math.*, **VIII,** 123–132 (1955).
15. K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
16. M. A. Crisfield, *Non-linear Finite Element Analysis of Solids and Structures*, Wiley, U.K., 1991.
17. T. J. R. Hughes, 'Numerical implementation of constitutive models: rate-independent deviatoric plasticity', in: S. Nemat-Nasser, R. J. Asaro and G. A. Hegemier (eds.), *Theoretical Foundation for Large-scale Computations for Non-linear Material Behavior*, Martinus Nijhoff Publishers, Dordrecht, 1984.

18. T. J. R. Hughes and J. Winget, 'Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis', *Int. J. Numer. Meth. Engng.*, **15,** 1862–1867 (1980).
19. S. W. Key and R. D. Krieg, 'On the numerical implementation of inelastic time dependent and time independent, finite strain constitutive equations in structural mechanics', *Comput. Methods Appl. Mech. Engng.*, **33,** 439–452 (1982).
20. J. C. Simo, 'A framework for finite strain elastoplasticity based on maximum plastic dissipation and the multiplicative decomposition. Part II: computational aspects', *Comput. Methods Appl. Mech. Engng.*, **68,** 1–31 (1988).
21. A. Vila, A. Rodríguez-Ferran and A. Huerta, 'A note on a numerical benchmark test: an axisymmetric shell under ring loads', *Commun. Numer. Meth. Engng.*, **13,** 181–192 (1997).